CERIAS Tech Report 2001-58

Agents and Protocols For Variable Information Assurance In Workflow Systems

Thomas Bellocci, Shimon Y. Nof
Center for Education and Research in
Information Assurance and Security
&
Department of Industrial Engineering
Purdue University, West Lafayette, IN 47907

Agents and Protocols For Variable Information Assurance In Workflow Systems

Thomas Bellocci and Shimon Y. Nof
Center for Education and Research
in Information Assurance and Security
and
Department of Industrial Engineering, Purdue University
West Lafayette, IN 47907-1398 USA

Abstract

The design and operation of autonomous agents to assure information in ERP systems of inter-networked enterprises are investigated. A variable information assurance implementation model is proposed based on the AIMIS model, and a risk assessment procedure is applied. The protocols and models needed to support variable assurance are introduced and their performance is assessed. Experimentation shows the possibility to reduce the processing time of requests without decreasing the proportion of trusted requests, compared to a systematic total assurance approach.

Keywords: Information assurance, Workflow systems, Autonomous agents, Protocols

1. INTRODUCTION

The development of inter-networked enterprises implies new requirements for the management of supply chains. Companies often have trouble obtaining valuable, timely information and exchanging correct data between different divisions. Previous research showed the critical importance of information assurance for inter-networked enterprises, and the need to automate the assurance practices as much as possible. The objective is to design information systems to automatically apply the assurance function, and not to expect the workers who interact with the system to include the assurance tasks as part of their job.

This article addresses the problem of how to design and operate agents to assure information in production enterprises. The context of information assurance in ERP systems is the focus of this research. Two questions are addressed in this article:

1) Can assurance flexibility be introduced in information processing without reducing the global level of the confidence of data?
2) What is the impact of autonomous agents' organization on the performance of the system and the functioning of a company?

The context of these two questions is the realization that in complex ERP systems, it is unrealistic to expect total assurance of all the data. In addition, it is assumed that information assurance is not a goal in itself, but a measure to improve the productive performance of companies.

2. LITERATURE REVIEW

2.1. Definition of information assurance

The research work, described by Bellocci and Nof (2001), explained the need of inter-networked companies to get quality data for managing their operations. As a result, information assurance was defined as the combination of: 1) Information security, 2)

Information integrity, and 3) Information significance. *Information security* means protecting information from malicious threats and damage due to external or internal sources. *Information integrity* should be understood as permanency of the information during communications and storage. Lastly, *information significance* refers to the value that the intended user can get out of the information when s/he receives it. The broader view considers assurance from the viewpoint of “quality assurance”. The broader definition is proposed as follows:

- Information assurance combines the requirements of information security, integrity and significance.
- Assuring information means having a safe information system, which guarantees that information is secure and at the same time keeps its integrity and its significance during its lifetime.
- The goal of information assurance is to provide trustworthy and significant information to users in operational, service systems that rely on the information for the fulfillment of their objectives.

2.2. Security and assurance agents

Autonomous agents system is a relatively recent research area. A comprehensive review and definition of agents have been available only recently (e.g., Franklin and Graesser, 1997; Nof, 1999). Also, distinctions between agents are only starting to appear.

Security agents have been among the first type of agents to be studied. An early implementation of security agents can be found in the work described by Crosbie and Spafford (1995). The authors describe an Intrusion Detection System, in which software agents are used to monitor potential security flaws. This research work provides all the necessary elements for designing an autonomous agent system. But in this case, security agents are out of the “production circle”.

A new step in the field of information assurance was made by Varadharajan et al. (1998). They describe a security agent-based distributed authorization system. In the prototype, “productive” software agents are provided with security features. For instance, when a customer logs on a bank ATM, a withdrawal agent is created and migrates to the bank host server to execute the transaction. To increase the level of assurance, the capabilities of the agent need to be restricted to ensure that the local resources of the host are protected from unauthorized actions by the agent. As a result, the agent, which contains the code of the action that it can perform, is also delegated some of the user's privileges and security characteristics that are required to perform these actions. In this example, assurance features have been added to agent, as the agent will carry with it the user's privileges list. Nevertheless, the "productive" task, money withdrawal, is still separated from the "assurance" task, namely checking agent's rights. A combination of tasks is not envisaged.

In conclusion, past research addressed security and assurance agents. But it never considered a combination of security or assurance tasks with production tasks in the same agent. The issue of tasks combination in agents for assurance purpose is addressed in this article.

2.3. Agent-based workflow system

An autonomous agent system architecture able to supervise processes has been described by Kim (1996), and Kim and Nof (2000). In these papers, the authors introduce the AIMIS (Agent-based Integration Model of Information Systems). This architecture (Figure 1) is composed of two types of agents: GCAs (Global Coordination Agents) and LCAs (Local Coordination Agents). GCAs reside in a central computer, and are again classified into three types of agents: triggering agents, execution agents, and coordination agents. A triggering agent g^T receives events from LCAs. It matches events with a process library, and triggers an execution agent, g_i^E, to supervise the identified process. The execution agent controls the execution sequence of component data activities of the process, which is represented by a graph called DAF-Net. It sends activity execution requests to appropriate LCAs and receives the execution results from the LCAs. Finally, the coordination agents control the execution sequence of multiple processes to prevent them from generating incorrect results. This autonomous agent model was used in this research work as a model of agent-based workflow system.

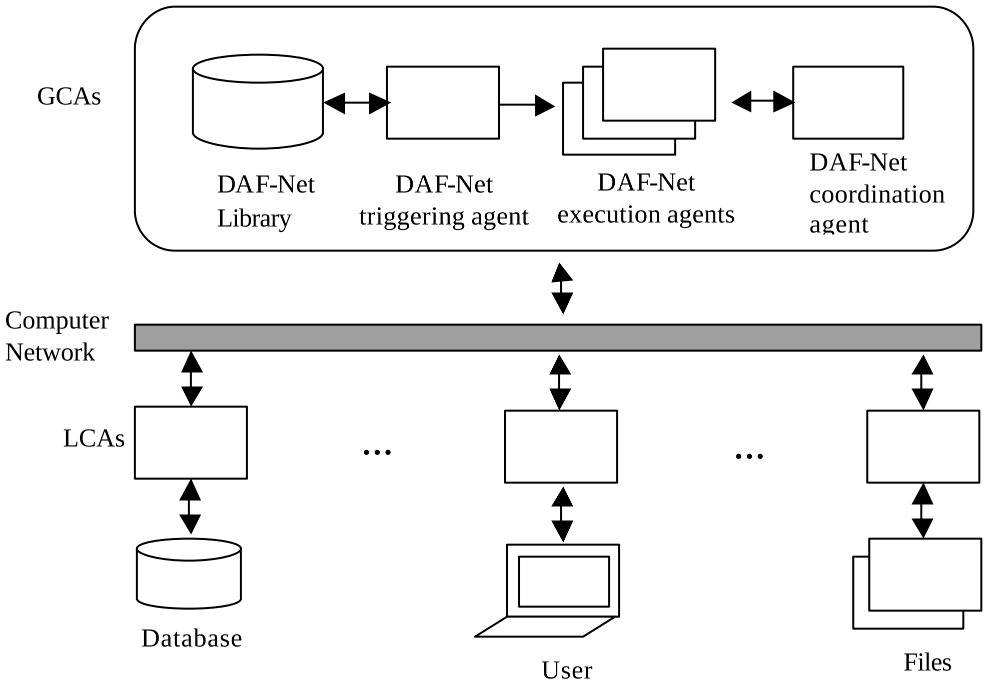

Figure 1. Agent-based Integration Model of Information Systems (Kim, 1996)

3. VARIABLE INFORMATION ASSURANCE WITH AGENTS

In an ERP system, autonomous agents can perform production-related tasks or assurance-related tasks. In the frame of this research, two dimensions of autonomous agent systems are investigated:

a. the conditional execution of assurance tasks, and

b. the agents used to perform the tasks.

The first problem is referred to as "variable assurance" problem, and the second one is called "task combination" problem. The first section of this chapter describes the justification for variable assurance and presents the basis for its implementation. The second section focuses on task combination in agents, and introduces models for assurance in autonomous agent systems.

3.1. Variable assurance

3.1.1. Introduction

In a distributed information system, several types of transactions can take place. Not all of the requests have the same importance for the functioning of the company. A set of lab experiments was conducted by Bellocci and Nof (2001) using an ERP simulator, called MICSS. These experiments demonstrated the differences of impact of information failures on transactions. Certain types of information failures are more critical for the company and, therefore, should be monitored first. The implementation of variable assurance requires two steps: 1) Evaluating the importance of performing assurance tasks for a given transaction, and 2) Deciding if the assurance tasks should be performed according to this importance level.

3.1.2. Risk assessment

The decision of whether or not to perform assurance tasks for a given production request needs to be supported by two separated information gathering activities:

a. The Request Analysis
b. The Context Analysis.

The purpose of the Request analysis is to gather the request's characteristics to tailor an assurance process to the request needs, based on the analysis of the critical information assurance failures presented Bellocci and Nof (2001).

A distributed information system is a dynamic entity with changing characteristics. Server utilization changes all the time, communications can slow down, intrusions can occur… The purpose of the Context Analysis is to gather information about the system to adjust the assurance processes to the status of the system. Autonomous agent systems for distributed system indicators monitoring were described in earlier in the literature (e.g., Crosbie and Spafford, 1995; De Meer et al., 1998). These systems show how to use autonomous agents to obtain a dynamic overview of the system status, including security weaknesses and data processing performance.

The functioning of the AIMIS described by Kim (1996) was briefly explained in Section 2. This model relies on processes defined in a library. A triggering agent g^T is responsible for process recognition and triggers an execution agent g_i^E to supervise the execution of the process i. This model serves as the basis for variable assurance implementation. Assurance Tasks are included in the existing processes of the library. The Assurance Tasks added to the processes take into consideration the process characteristics and risk of information assurance failures. The conditional execution of assurance tasks works as follows: g_i^E reads the next task of the process. If it is an Assurance Task, g_i^E analyzes the result of the Request Analysis and the result of the Context Analysis to decide if it should trigger the execution of the Assurance Task, according to a given Variable Assurance Protocol V.

The objective of the autonomous agent system is to increase the Assurance Level $\alpha(R_i)$ attributed to a request R_i after the risk assessment procedure to the minimum Assurance Level L(t) required at time t by the system by triggering the execution of appropriate assurance tasks (Figure 2).

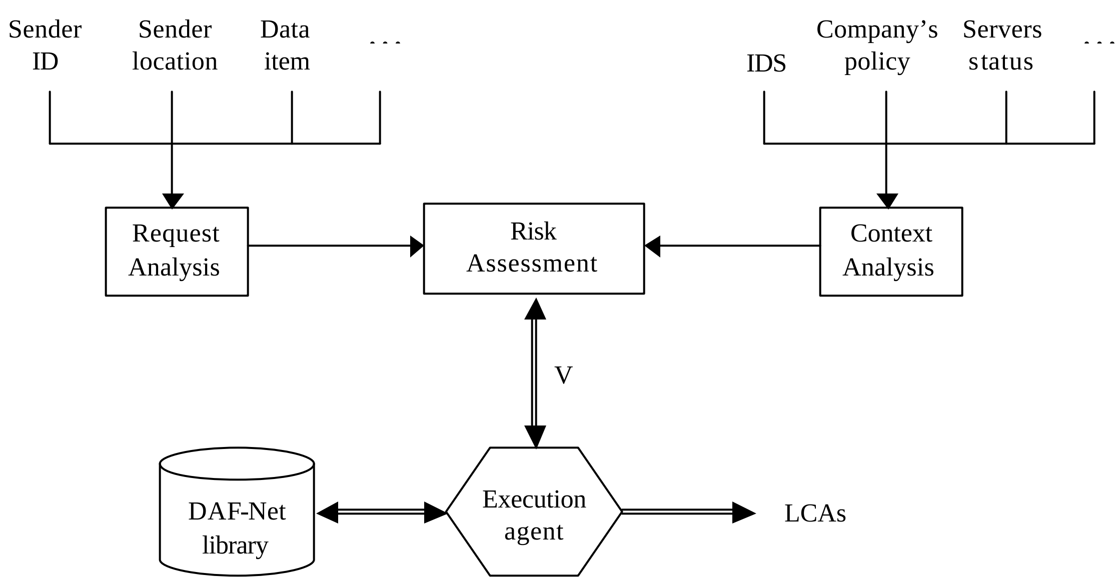

Figure 2. Variable information assurance implementation model

3.1.3. Variable Assurance Protocols

Three different variable assurance protocols were designed to support the implementation of the variable assurance model. The results of the MICSS lab experiments showed the importance to consider two different request characteristics:

(1) The assurance needs of the request, that can be assimilated to the need of assurance features (trustworthiness, completeness, integrity…) for instance due to the information sender location, or receiver identity, and

(2) The priority of the request, that corresponds with the need to receive the information on time.

Based on this conclusion, three different Variable Assurance Protocols with different logic were designed:

a. VAP0 assures all the requests,

b. VAP1 assures requests based on their assurance needs, and

c. VAP2 assures requests based on their assurance needs and priority level,

These protocols are described below. They are modeled and analyzed later, in Section 4.

VAP0 (Total assurance):

FOR ANY R_i

go to execute assurance task

VAP1 (Needs-based assurance):

IF (R_i assurance needs are high or low)

THEN go to execute assurance task

ELSE go directly to execute production task

VAP2 (Needs- and priority-based assurance):

IF (R_i priority is low) OR (R_i assurance needs are high)

THEN go to execute assurance task

ELSE go directly to execute production task

3.2. Assurance models:

In the frame of this research, two categories of agents were considered: 1) Dedicated agents, A_D, and 2) Polyvalent agents, A_{AP}. There are two types of dedicated agents: 1) Assurance dedicated agents, A_A, that can only perform Assurance Tasks, T_A, and 2) Production dedicated agents, A_P, that can only perform Production Tasks, T_P. A polyvalent agent, A_{AP}, is able to execute both a production task and the associated assurance task. The execution of these two tasks in a row by the same agent is referred to as an assurance-production task, and noted T_{AP}. A polyvalent agent can also execute single production tasks, T_P, by skipping the assurance part of its code.

Following these observations, three Assurance Models were proposed depending on the agents available to execute the Assurance and Production Tasks, namely: 1) The Separated Model, M_{Sep}, 2) the Combined Model, M_{Com}, and 3) the Mixed Model, M_{Mix}. The nature of the agents involved in each model is summarized in Table 1.

Table 1. Summary of Assurance Models

	Type of agents involved	
Model	$\mathbf{A_D}$	$\mathbf{A_{AP}}$
M_{Sep}	Yes	No
M_{Com}	No	Yes
M_{Mix}	Yes	Yes

4. EXPERIMENTATION

4.1. System's description

Requests enter the agent-based workflow system with an inter-arrival time exponentially distributed with a mean of λ. A request is composed of an assurance and a production part, and has specific assurance needs that can be represented by an Assurance Level $\alpha(R_i)$. The company decided of an Assurance Policy that requires a request to reach the Assurance Level $L(t)$ to be trusted. The objective of the autonomous agent system is to increase the Assurance Level $\alpha(R_i)$ of a request R_i entering the system to the minimum Assurance Level $L(t)$ required at time t by the system.

Given $\alpha(R_i)$ and $L(t)$, the triggering agent decides whether or not R_i needs to be assured prior to the execution of the production part of the request, according to a variable assurance protocol V. Different types of agents are available for task execution in the assurance model M. When a request needs to be assured, the triggering allocate the task execution to agents using a communication protocol CP(A). An assurance task performed by a Polyvalent agent, A_{AP}, increases the assurance level of a request 1.2 times more than an assurance task performed by an Assurance dedicated agent, A_A, following considerations about the risk of failures during task execution and agent migration. When a request is limited to production task, the triggering allocate the task execution to agents using a communication protocol CP(P).

As a consequence, an autonomous agent system S for workflow monitoring can be represented by a 3-tuple S = (M, V, EP). M is the assurance model used to handle assurance tasks in the system. V is the protocol used to distinguish between the production planning requests that need to be assured and the ones that do not need to be assured. Finally, the autonomous agent system S works under the environmental parameters EP, represented by a 5-tuple, EP = (λ, CP(A), CP(P), L, N). The inter-arrival time of production planning requests is exponentially distributed with a mean of λ. CP(A) is the communication protocol to select the agent in charge of the assurance part of a request, and CP(P) the protocol to select the agent in charge of the production part of a request. Assuming a level of assurance is required, L represents the company's assurance policy level. N is the total number of agents available in the system. For simplicity, it is assumed that the assurance policy level L is fixed over time.

<u>Example:</u> Autonomous agent system S = [M_{Sep}; VAP1; (0.5, CP_1, CP_2, 10, 500)]

In this example, the autonomous agent system involves only dedicated agents. The selective execution of assurance tasks is determined by VAP1, i.e. on the basis of requests' assurance needs. The arrival of requests is exponentially distributed with a mean of 0.5 seconds. The allocation of assurance tasks to agents is determined using protocol CP_1. The allocation of production tasks to agents is determined using protocol CP_2. Ten agents are available in the system, and the company has fixed the assurance policy level to 500 A.U. (Assurance Units).

4.2. Simulation model

The logic of the simulation model is presented in Figure 3.

The time between request arrivals is exponentially distributed with a mean of λ. Each request has a specific Priority Level $p(R_i)$. Differences in request priorities are modeled using a uniform distribution. A request can have a priority equals to 1(high), 2(medium) or 3(low).

Each request arrives with a specific Assurance Level $\alpha(R_i)$. Differences in request assurance needs are modeled using a normal distribution Normal(μ_{AL}, σ_{AL}) for $\alpha(R_i)$. To trigger Assurance Tasks we will distinguish between requests that have no assurance needs, low assurance needs or high assurance needs using the company Assurance Policy L (Figure 4).

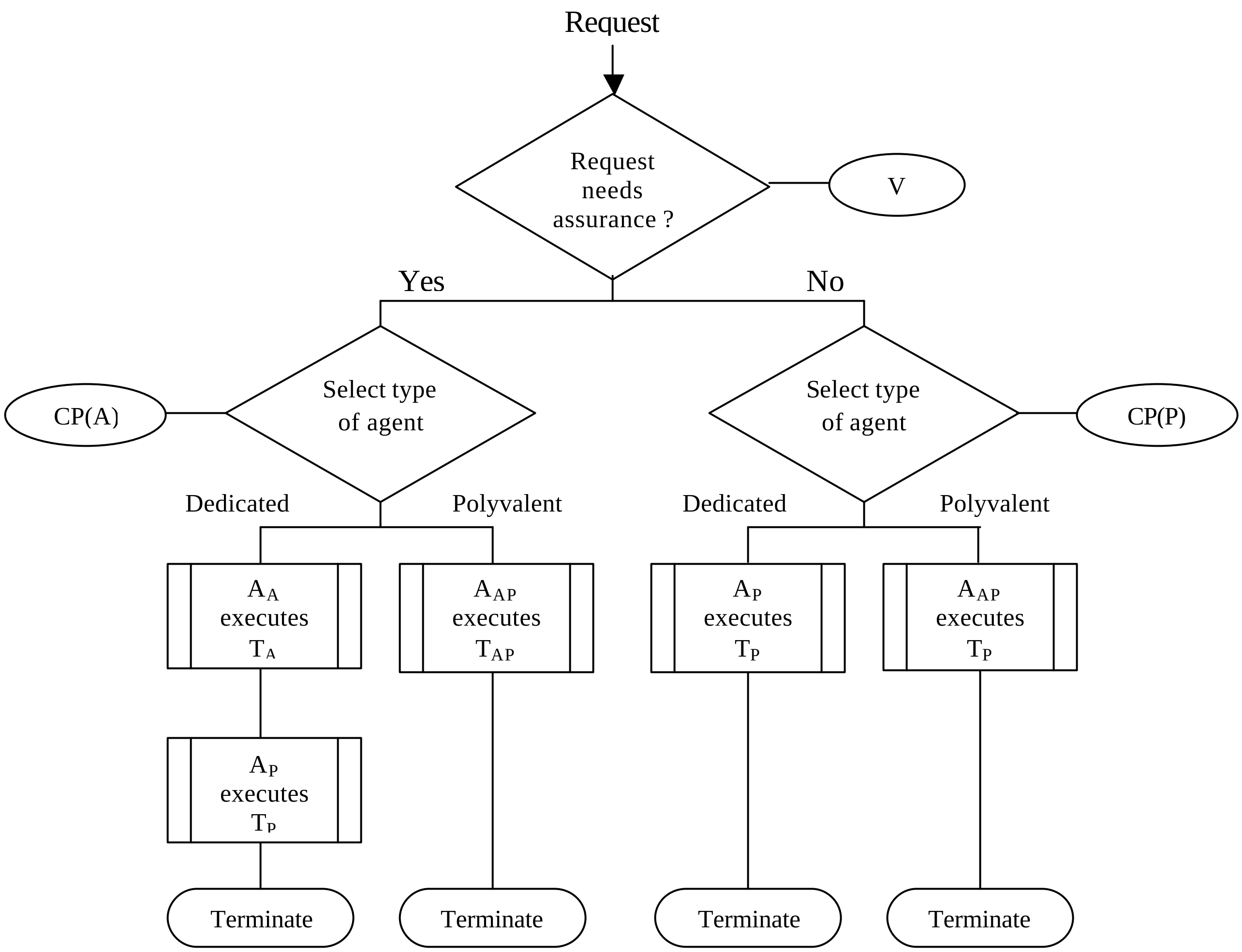

Figure 3. Logical chart of the simulation model

The execution agent g_i^E, responsible of monitoring the execution of request R_i, selects which requests should be assured using a Variable Assurance Protocol, V. The three protocols studied in this research were presented in Section 3.1.

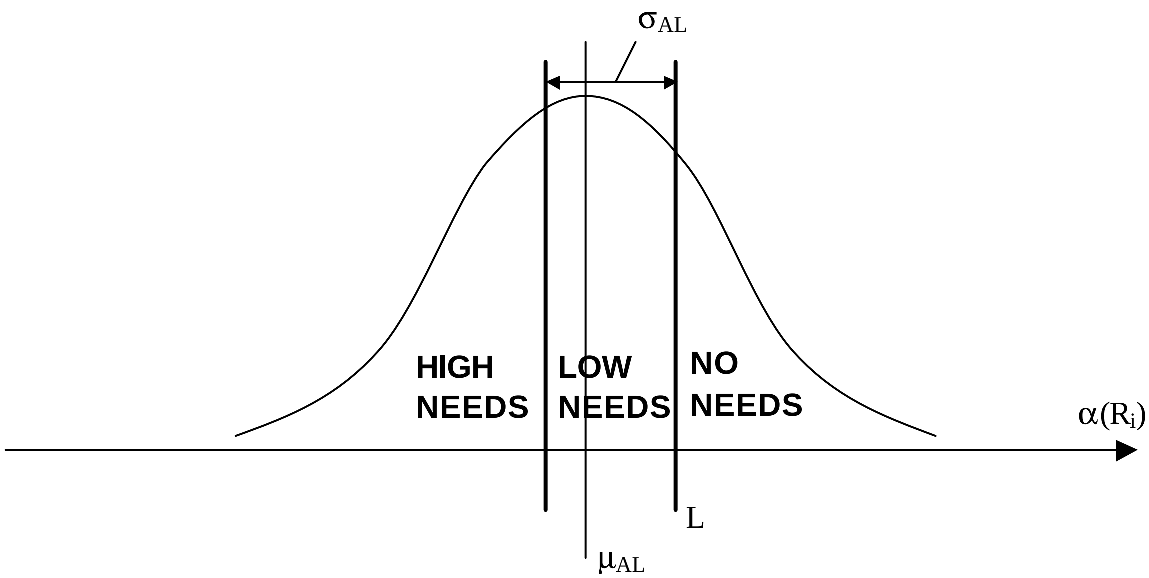

Figure 4. Definition of assurance needs using request assurance level and assurance policy

The execution agent g_i^E, responsible of monitoring the execution of request R_i, selects which requests should be assured using a Variable Assurance Protocol, V. The three protocols studied in this research were presented in Section 3.1.

If g_i^E decides that the request R_i must be assured, it will look for an agent to perform the Assurance task or the Assurance-Production task. The execution agent uses the task allocation protocol CP1 for this purpose. It is defined as follows:

CP1:

IF (no agents A_A in the system) THEN send R_i to agent A_{AP}

ELSE IF (no agents A_{AP} in the system) THEN send R_i to agent A_A

ELSE

IF (R_i assurance needs are high) THEN send R_i to agent A_{AP}

ELSE send R_i to agent A_A

g_i^E selects the agent responsible for the production task of request R_i using the task allocation protocol CP2, defined below:

CP2:

IF (no agents A_P in the system) THEN send request to agent A_{AP}

ELSE IF (no agents A_{AP} in the system) THEN send request to agent A_P

ELSE

Balance (Queue for agent A_P) and (Queue for agent A_{AP})

IF (Queue for agent A_P is smaller) THEN send to agent A_P

ELSE send to agent A_{AP}

4.3. Experimental questions

The following experimental research questions were investigated to answer the research problems formulated in Section 1, namely: "Can flexibility be introduced in information processing without reducing the global confidence level of data?" and "What is the impact of autonomous agents organization on the performance of an autonomous agent system?"

1) Research Question 1:
   What are the significant parameters for the processing time of requests?
2) Research Question 2:
   What are the significant parameters for the exit assurance level of requests?
3) Research Question 3:
   What is the best variable assurance protocol overall?
4) Research Question 4:
   What is the best assurance model overall?
5) Research Question 5:
   What is the best combination of Variable Assurance Protocol and Assurance Model given an assurance policy level and a number of agents?

4.4. Design of experiment

Two metrics were used to assess the performance of an autonomous agent system S:

(1) The processing time of requests, θ (S)

(2) The assurance exit level of requests, η(S),

Four independent variables were used:

(1) Variable assurance protocol, symbolized V, with three levels:

   a. V0 = VAP0 (Total assurance)
   b. V1 = VAP1 (Needs-based assurance)
   c. V2 = VAP2 (Needs- and priority-based assurance)

(2) Assurance model, symbolized M, with three levels:

   a. M1 = M_{Sep} (Separated model)
   b. M2 = M_{Com} (Combined model)
   c. M3 = M_{Mix} (Mixed model)

(3) Assurance policy level, symbolized L, with three levels:

   a. L1 = 300 A.U. (Low requirements)
   b. L2 = 500 A.U. (Medium requirements)
   c. L3 = 700 A.U. (High requirements)

(4) Total number of agents, symbolized N, with three levels:

   a. N1 = 10 agents (Low quantity)
   b. N2 = 15 agents (Medium quantity)
   c. N3 = 20 agents (High quantity)

Based on this design of experiment, 81 different treatments were simulated. For each treatment, two simulation runs were executed with different random numbers. During a run, the processing time and exit assurance level of the first 500 executed requests were recorded. The stationary state is reached after 20 to 50 requests, depending on the treatment. As the transient regime ends relatively quickly, the first requests were kept in the pool of 500 requests used for the analysis.

4.5. Results and conclusions

The answers to the experimental research questions formulated in Section 4.3. are presented below:

1) *What are the significant parameters for the processing time of requests?*

According to the ANOVA results, all of the four parameters V, M, L, N and their interactions are significant with a confidence level of 95%. Hence:

(E1) $\theta(S) = f(V, M, V*M, L, V*L, M*L, V*M*L, N, V*N, M*N, V*M*N, L*N, V*L*N, M*L*N, V*M*L*N)$

As a conclusion, Variable Assurance Protocols and Assurance Models have a significant impact on the processing time of requests.

2) *What are the significant parameters for the exit assurance level of requests?*

According to the ANOVA results, only some of the parameters have significant impact on the exit assurance level of the request with a confidence level of 95%. In fact:

(E2) $\eta(S) = f(V, M, V*M, L, V*L, M*L, V*M*L)$

As a conclusion, Variable Assurance Protocols and Assurance Models have a significant impact on the exit assurance level of requests.

3) *What is the best variable assurance protocol overall?*

Decision-makers relying on information to complete their tasks are particularly interested in the proportion $\tau(S)$ of trusted requests that exit the system S. A request is called "trusted" if its exit assurance level is higher than the assurance policy level of the company.

A Student-Newman-Keuls range test was used to rank the variable assurance protocols with a confidence level of 95%. The results are as follows:

a. The processing times given by the Variable Assurance Protocols are all significantly different, and overall the protocols can be ranked by increasing processing time: VAP2 < VAP1 < VAP0
b. The following ranking appears regarding the proportion of trusted requests given by the protocols: (VAP1 = VAP0) > VAP2

As a consequence, two protocols provide interesting results. Both VAP1 and VAP2 offer a reduction of processing time compared to VAP0 (Figure 5). The needs-based protocol VAP1 does not decrease the proportion of trusted requests. The needs- and priority based protocol VAP2 allows a larger reduction of processing time than VAP1, but also implies a diminution of the proportion of trusted requests (Figure 6).

4) What is the best Assurance Model overall?

A Student-Newman-Keuls range test was used to rank the Assurance Models with a confidence level of 95%. The results are as follows:

a. The processing times given by the Assurance Models are all significantly different, and overall the models can be ranked by increasing processing time: $M_{Com} < M_{Mix} < M_{Sep}$

b. The following ranking appears regarding the proportion of trusted requests given by the assurance models: $(M_{Com} = M_{Mix}) > M_{Sep}$

It can be concluded that the Combined assurance model M_{Com} performs better than the Separated and Mixed models. It is the fastest model in requests processing (Figure 7), and provides the largest proportion of trusted requests (Figure 8).

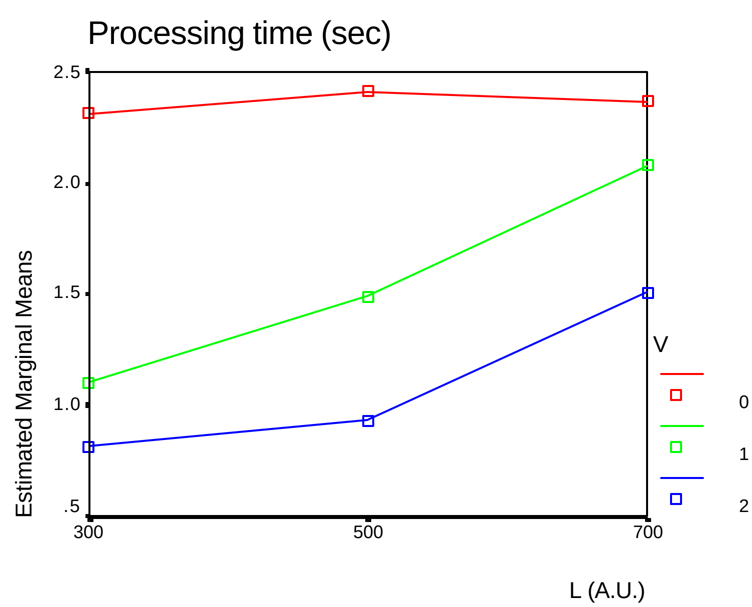

Figure 5. Mean processing time of requests for different variable assurance protocols in function of the assurance policy level.

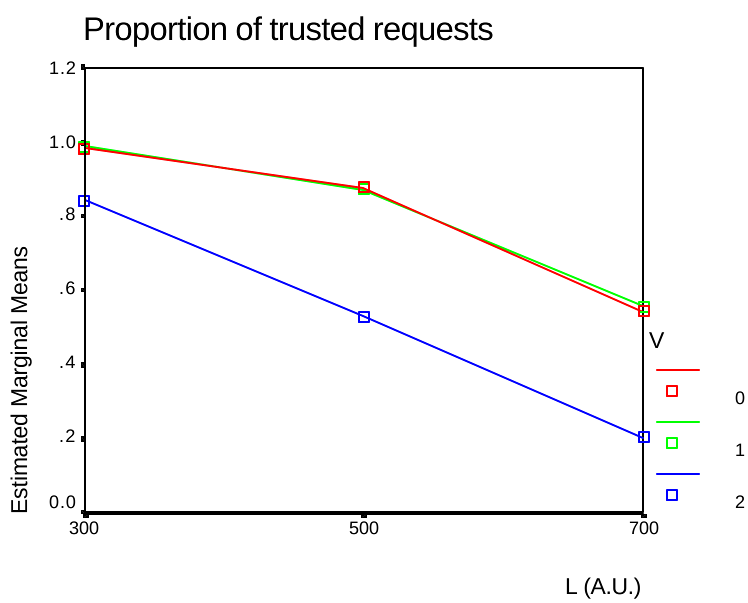

Figure 6. Proportion of trusted requests for different variable assurance protocols in function of the assurance policy level.

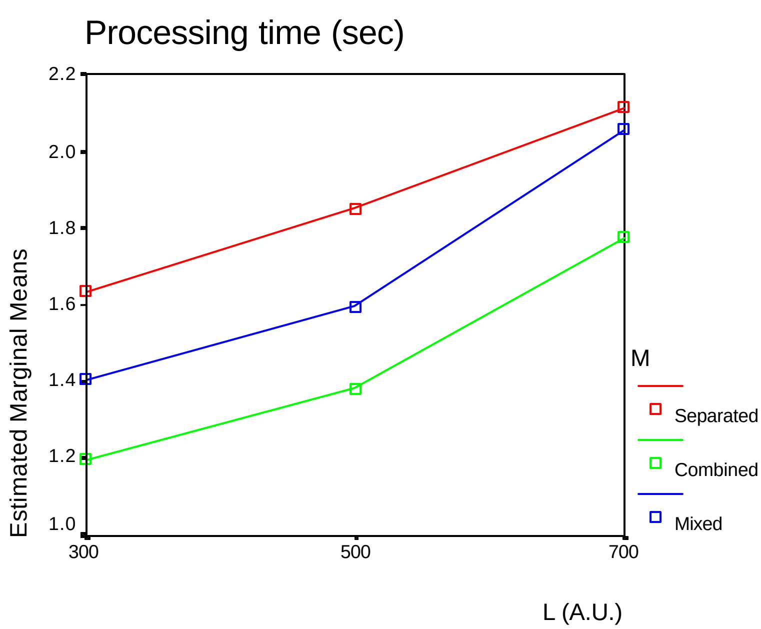

Figure 7. Mean processing time of requests for different assurance models in function of the assurance policy level.

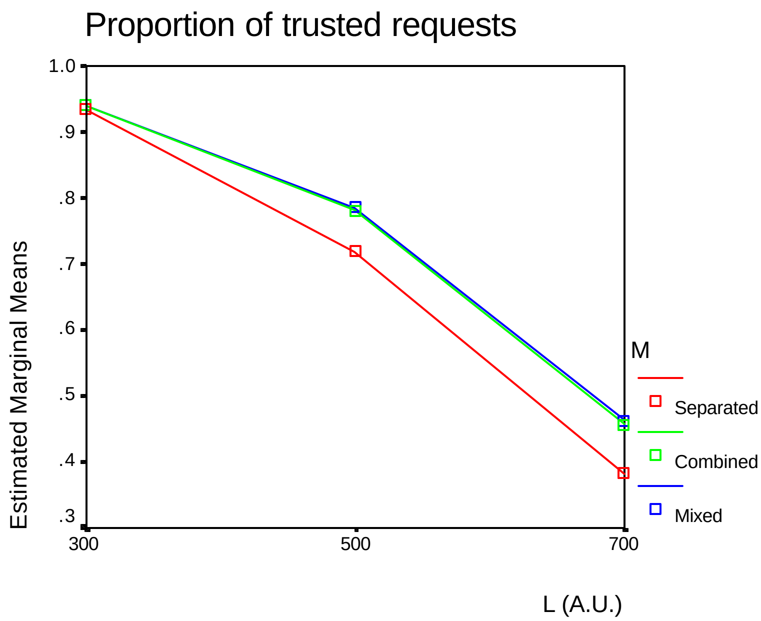

Figure 8. Proportion of trusted requests for different assurance models in function of the assurance policy level.

5) *What is the best combination of Variable Assurance Protocol and Assurance Model given a combination of Assurance Policy Level and Number of Agents?*

Selecting a combination of Variable Assurance protocol and Assurance Model is a trade-off between low processing time of requests, and high proportion of trusted request. Two treatments can compete with the Total Assurance protocol: VAP1*M_{Com} and VAP2*M_{Com}. Their performances compared to Total Assurance are summarized in Tables 2 and 3, regarding the mean processing time of requests, $\theta(S)$, and the proportion of trusted requests, $\tau(S)$.

Both VAP1*M_{Com} and VAP2*M_{Com} are interesting alternatives to Total assurance. VAP2, however, implies a diminution of the proportion of trusted requests. When the company's assurance requirements are low (i.e., L = 300 A.U.), the reduction of $\tau(S)$ is limited. The conclusions about which combination of Variable Assurance

Protocol and Assurance Model to choose for a given combination of L and N are summarized in Table 4.

Table 2. Comparison of VAP1*M_{Com} performance to Total Assurance

VAP1*M_{Com} versus Best VAP0		**Assurance Policy Level**		
		300 A.U. (low requirements)	**500 A.U.** (medium requirements)	**700 A.U.** (high requirements)
Total Number of Agents	**10** (low)	θ(S): -50% τ(S): -0%	θ(S): -60% τ(S): -0%	θ(S): -34% τ(S): -0%
	15 (medium)	θ(S): -72% τ(S): -0%	θ(S): -45% τ(S): -0%	θ(S): -19% τ(S): -0%
	20 (large)	θ(S): -70% τ(S): -0%	θ(S): -44% τ(S): -0%	θ(S): -20% τ(S): -0%

Table 3. Comparison of VAP2*M_{Com} performance to Total Assurance

VAP1*M_{Com} versus Best VAP0		**Assurance Policy Level**		
		300 A.U. (low requirements)	**500 A.U.** (medium requirements)	**700 A.U.** (high requirements)
Total Number of Agents	**10** (low)	θ(S): -69% τ(S): -16%	θ(S): -79% τ(S): -38%	θ(S): -55% τ(S): -65%
	15 (medium)	θ(S): -82% τ(S): -15%	θ(S): -72% τ(S): -43%	θ(S): -46% τ(S): -59%
	20 (large)	θ(S): -80% τ(S): -15%	θ(S): -71% τ(S): -37%	θ(S): -44% τ(S): -64%

Table 4. Best combination of Variable Assurance and Assurance Model depending on the Assurance Policy Level, and Number of Agents

		Assurance Policy Level		
		300 A.U. (low requirements)	**500 A.U.** (medium requirements)	**700 A.U.** (high requirements)
Total Number of Agents	**10** (low)	VAP2*M_{Com} (for time) VAP1* M_{Com} (for assurance)	VAP1* M_{Com}	VAP1* M_{Com}
	15 (medium)	VAP2*M_{Com} (for time) VAP1* M_{Com} (for assurance)	VAP1* M_{Com}	VAP1* M_{Com}

	20 (large)	VAP2*M_{Com} (for time) VAP1* M_{Com} (for assurance)	VAP1* M_{Com}	VAP1* M_{Com}

4.6. Validation of experiments

The simulation experiments need to be compared to known results to be validated. The influence of the parameters has been investigated independently. It appears that the processing time increases when the number of agents decreases. When the assurance policy level increases, the processing time increases because the number of assurance tasks to be performed increases. These simple observations validate the correct behavior of the simulation model from a practical point of view.

An industry survey was conducted and presented by Bellocci and Nof (2001). The insights coming from the analysis of the survey can be used to validate our experiments from the corporate viewpoint. For instance, managers explain in the survey that no company ever reaches a proportion of trusted requests equal to 100%. Also, when the assurance policy level of the company increases fewer requests can meet the requirements, and the proportion of trusted requests decrease. These observations validate the behavior of the simulation model.

5. SUMMARY AND DISCUSSION

5.1. Summary

The analysis of the experiments showed that flexibility can be introduced in assurance tasks execution without reducing the confidence level of data. The Total assurance protocol VAP0 provides the best exit assurance level of requests, but can overshoot the assurance level required by the company's assurance policy. The requests executed using protocol VAP1 (needs-based assurance) exit the system with a significantly smaller assurance level than with VAP0, at a confidence level of 95%. Nevertheless, the proportion of trusted requests is similar with VAP1 and VAP0, and the processing time with VAP1 is significantly smaller than with VAP0, at a confidence level of 95%. As a consequence, flexibility in execution of assurance tasks can be introduced in agent-based workflow system using protocol VAP1, which allows the system to reach similar confidence level to total assurance and save significant processing time.

The results of the experiments showed that in the case where assurance tasks are serialized with production tasks, the best assurance model is the one involving only polyvalent agents, M_{Com}. Compared to M_{Sep} and M_{Mix}, this model reaches indeed the smallest processing time and highest proportion of trusted requests for any assurance policy or number of agents.

The best combination of variable assurance protocol and assurance model depends essentially on the company's assurance policy level. When the requirements are medium or high, VAP1 combined with M_{Com} allows a significant reduction of the processing time compared to Total Assurance without reducing the proportion of trusted requests. When the requirements are low, companies can decide between using VAP1* M_{Com}, that reduces the processing time without decreasing the confidence level, and VAP2*M_{Com}, that implies a larger processing time reduction than VAP1 with however a decrease of 15% in the proportion of trusted requests. In this case, information system managers have to decide what is the best trade-off for the functioning of the company.

5.2. Future research work

The following directions can be recommended for future research:

(1) It has been assumed in the simulation models that the assurance policy level was fixed over time. The influence of assurance policy level variation over time could be investigated.

(2) In this research work, it has been assumed that the entry assurance level of the requests could be modeled by a normal distribution. A possible direction for future research would be to study if another law could fit better the actual distribution of requests' entry assurance level, and assess the performance of Variable Assurance Protocols and Assurance Models with this new distribution.

(3) The simulation models focused on the sequence of optional assurance tasks followed by production tasks. In such a case, the Combined assurance model appeared to be the most advantageous. The development of assurance protocols to distinguish between the processing of requests that need parallel assurance tasks or the ones that need serial assurance tasks could be investigated.

(4) The variable assurance approach presented in this research work showed that significant resources can be saved by adjusting the assurance tasks to the request and the context. However, additional resources can be saved if the assurance tasks that are performed on concurrent requests are taken into account, as they increase indirectly the assurance level of the given request. Negotiation-based variable assurance protocols could be investigated to solve this research problem.

6. ACKNOWLEDGEMENTS

This research was supported by the CERIAS (Center for Education and Research in Information Assurance and Security) at Purdue University.

REFERENCES

Bellocci, T., and Nof, S.Y., Context of Information Assurance in Inter-Networked Enterprises, *CERIAS Technical Report 2001-58, Research Memorandum No. 01-19, School of Industrial Engineering Purdue University*, December 2001.

Crosbie, M., and Spafford, E., Active Defense of a Computer System using Autonomous Agents, *COAST Technical Report 1995-008, Purdue University*, 1995.

De Meer, H., Puliafito, A., and Tomarchio, O., Management of Quality of Service with Software Agents, *Cybernetics and Systems*, v 29, n 5, 1998, 499-523.

Franklin, S., and Graesser, A., Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents, *Intelligent Agents III, Muller, J.P., Wooldridge, M.J., and Jennings, N.R., (ed.)*, 1997, 21-35.

Kim, C.O., DAF-Net and Multi-Agent Based Integration Approach for Heterogeneous CIM Information Systems, *Purdue University, School of Industrial Engineering, Ph.D. Dissertation*, August 1996.

Kim, C.O., and Nof, S.Y., Investigation of PVM for the Emulation and Simulation of a Distributed CIM Workflow System, *International Journal of Computer Integrated Manufacturing*, v 13, n 5, 2000, 401-409.

Nof, S.Y., Intelligent, Collaborative Agents, *Research Memorandum No. 99-08*, *School of Industrial Engineering, Purdue University,* March 1999.

Varadharajan, V., Kumar, N., and Mu, Y., Security Agent Based Distributed Authorization: An Approach, *21st NISSC Proceedings*, Crystal City, VA, October 1998.