

Intelligent Decision Support for Architecture and Integration of Next Generation Enterprises

Amjad Umar
 Information and Communication Systems, Schools of Business
 Fordham University, New York, USA
 E-mail: umar@amjadumar.com, website: amjadumar.com

Keywords: enterprise architectures, enterprise integration, next generation enterprises, computer aided decision support, PISA, business patterns

Received: April 24, 2007

Architectures and integration of emerging next generation enterprises (NGEs) require a series of complex decisions. This paper describes an intelligent decision support environment that uses patterns, best practices, inferences, and collaboration for enterprise architecture and integration projects. This environment consists of a set of intelligent advisors that collaborate with each other in a fashion similar to a team of consultants who are working on an integration project. It guides the user to appropriate strategic choices, architectural configurations, COTS (commercial off the shelf) packages and project plans. Instead of rushing to automatic code generation from business process models, this paper takes a more cautious approach that is based on practical experience and first concentrates on a decision support environment that will introduce more automation in later iterations.

Povzetek: Opisano je okolje za integracijo projektov z uporabo najboljših praks.

1 Introduction

Enterprise architecture and integration projects are complex undertakings especially in the emerging next generation enterprises (NGEs) that rely on deep technology stacks on a daily basis. Specifically, NGEs rely on web-technologies, mobile services, real-time business activity monitoring, agility, self-service, and widely distributed operations to conduct business [31]. Modern architecture and integration projects (AIPs) require participation of, and information sharing between, IT staff, IT managers, consultants, customers, and even business partners. Based on lessons learned from several industrial consultation and academic/research assignments, and review of vendor products and research efforts, we have found that comprehensive decision support environments are needed to lead the participants systematically through the maze of business scenarios, strategic choices involving outsourcing and warehousing, and integration tradeoffs based on cost, performance and security issues. Although environments of this nature are urgently needed, they are virtually non-existing. To fill this gap, we have initiated research on decision support for enterprise integration projects with the following “requirements”:

- **Support different players of the integration projects.** These projects require many decisions that need to be shared, monitored and controlled by different parties. Thus it is important to capture high level business process models and enterprise ontologies for ease of communications, support different views and what-if scenarios for different project

participants, capture different architectural configurations (e.g., outsourcing and data warehousing) that impact integration solutions, and facilitate evaluation of cost, performance and security tradeoffs before implementation.

- **Adopt a breadth first approach.** Development of an environment that supports decisions in different phases of a project should be a *parallel* effort and not an afterthought once all the individual problems have been solved. It is important to provide visibility throughout an integration project as it proceeds through various stages of its life cycle. This is especially crucial for IT managers because they need a total project view.
- **Approach automation systematically.** Automation of integration projects is a desirable goal but it is best to take a cautious approach that first concentrates on capturing/managing knowledge and supporting decisions throughout the project. This will help us better understand what to automate and when, instead of quickly automating the irrelevant activities or generating code from business process models without considering architectural details.
- **Develop a set of collaborating advisors.** Instead of one expert system, it is best to develop a set of intelligent advisors that collaborate with each other in a fashion similar to a team of consultants who are working with each other on real-life integration projects. These advisors should use *patterns* to capture the common and best practices instead of every possible point in the

solution space, heavily rely on *inferences* to reach conclusions instead of asking too many irrelevant questions, and *collaborate* with other consultants to solve complex problems.

- **Educate the users.** Due to the complexity and recurring nature of integration projects, the environment must support e-learning through online tutorials, guides, explanations, and justifications.
- **Stay close to standards and industrial developments.** The environment must closely follow Web Services and results of other efforts such as the INTEROP, Integration Consortium, and OMG (Object Management Group).

We have developed PISA (Planning, Integration, Security, and Administration) environment, an intelligent decision support system that addresses these issues. PISA consists of a set of collaborating advisors that are segmented into two major modules: a) PlanIT (Planner for IT), discussed elsewhere [38], that concentrates on IT planning projects and develops a plan at enterprise level, and b) Architecture and Integration Module (AIM), the focus of this paper, that deals with the architecture and integration issues. Section 2 describes the conceptual model at the core of AIM and Section 3 illustrates AIM through a simple example. PISA is discussed in [40].

2 Decision Support Model for Enterprise Architecture and Integration

Figure 1 displays the proposed decision support model for architecture and integration projects (AIPs) of NGEs. This simple model concentrates on four broad stages (enterprise modeling, requirements development, architecture selection, and solution evaluation) and has several important features. First, it specifically addresses

the decision support issues raised previously and also clearly captures the key decisions such as opportunity evaluation and BPO (Business Process Outsourcing), selection of applications and requirements for integration, integrated architecture choices (strategic, technology selection, architectural configuration), and integrated solution evaluation based on cost, performance and security. Second, the model is asynchronous – it captures the real life situation where different activities are typically invoked whenever enough input is available. Third, this model is intelligent because inferences are used heavily between the stages, and the planning knowledgebase provides extensive patterns and COTS (Commercial-Off-The-Shelf) information (explained later). In fact, the user inputs are optional in most cases. Thus this model can produce a quick sketch based on patterns and accumulated knowledge without any user interaction. Fourth, the plans are developed gradually in different stages and captured in the knowledgebase, thus later stages can learn from previous decisions. Fifth, this model conforms to the latest thinking in architectures, especially Model Driven Architecture (www.omg.org), which emphasizes the separation and isolation of platform specific issues as much as possible. For example, the enterprise modeling stage is completely independent of platform considerations – these considerations are introduced gradually in later stages. Finally, this model is specifically designed for a computer aided platform where each stage can be supported by an automated consultant (“advisor”), thus different artifacts can be introduced in each advisor and the advisors can collaborate with each other as a team of consultants to solve complex problems (Section 3).

The stages of this model, discussed below, are designed to address the decision support issues raised previously. A more formal treatment of this model with theoretical foundation is presented in Appendix A.

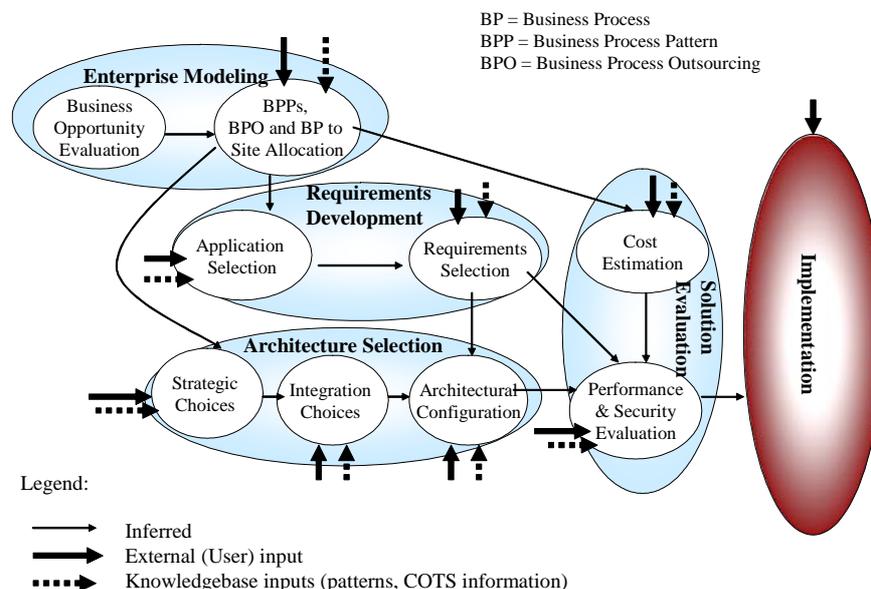


Figure 1: Decision Support Model for Enterprise Architecture and Integration

2.1 Stage 1: Enterprise Modelling

The main challenge in this stage is how to quickly build model of an enterprise. We have found that business process patterns (BPPs), shown in **Figure 2**, is a good starting point and of great practical value to identify business scenarios that drive integration projects and to conduct quick sensitivity analysis. For example, a business scenario may be concerned with re-engineering of one business process (e.g., purchasing), revamping of all applications in one functional area (e.g., back-office operations), a combination of the two, or an enterprise application integration (EAI) that encompasses all business processes (BPs) in all areas. **Figure 2** is based on an extensive review of enterprise ontologies [10, 21], business patterns [1, 15, 16] and industrial classifications (e.g., SAP’s Business Maps – www.sap.com). We have mapped this pattern to XML to facilitate high level sensitivity analysis of scenarios such as the following: a) if one BP is eliminated, then what other BPs will be impacted, b) if an application package that supports a BP is replaced with another application, what other applications/BPs will be impacted, c) which application, if replaced, will have the most impact in terms of integration, d) which application, if replaced, will have the least impact in terms of integration. We have created enterprise business patterns of this type for 9 industry segments that include manufacturing, healthcare, telecom, and others; and have mapped them to BPEL representation. We will investigate the use of business processing modeling languages such as BPML, UML 2.0, and others [7, 39] to represent these patterns.

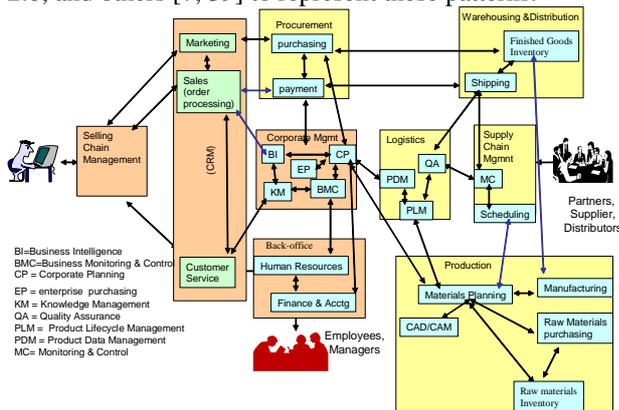


Figure 2: Business Process Pattern (BPP) for a Manufacturing Company

2.2 Stage 2: Requirements Development

Requirements development and analysis for integration projects can be considerably expedited through requirements templates that are based on a combination of requirements patterns suggested by Haage and Lappe [12], Ferdinandi [9], and standards

bodies [41]. This template should then be customized by considering factors such as user access devices, back-end apps, B2B apps, transaction value, transaction volume, number of partners, mobility support needed, etc. Several rules can be used to infer functional, interface and integration, mobility, performance and security requirements. For example, transaction volume can impact performance and transaction value can impact security requirements.

2.3 Stage 3: Architecture Selection

A three step process shown in **Figure 3** is proposed to help a user to explore various architectural strategies and then develop the integration components (front-end, back-end) of each architectural configuration. This approach generates many more architecture and integration patterns than have been reported in the literature [6, 11, 13]. The first step allows the user to choose between the following strategies for the *target applications* (applications of concern within an integration project):

- **Outsourcing (remote hosting):** decide where the target applications will reside: customer (your) site, service provider site, or a mixture.
- **Access in Place:** integrate without modifying any applications. Just access them by using adapters/mediators.
- **Data Warehouse:** build a "shadow" system to house the frequently accessed data. This is especially useful for BI (Business Intelligence) applications.
- **Migration:** re-architect and transition the target applications gradually

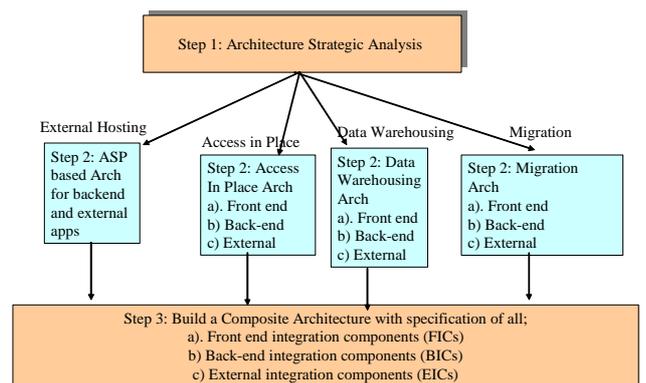


Figure 3: Architecture Steps

The architectures and the technologies needed to integrate these architectures depend on the strategic options of hosting, data warehouses, access in place, and migration. To illustrate these options and their impact on integration, let us consider a situation where a company XCorp chooses to outsource (rent) an online purchasing (OP) system from an ASP (e.g., use Amazon.com’s purchasing system through XCorp) but the inventory and shipping reside at XCorp site. In this

case, remote integration between XCorp applications and ASP is needed. In addition, the OP at the ASP site needs to integrate with the shipping and inventory apps at XCorp. This type of architecture raises interesting questions about security, performance, and interoperability. These questions are not raised if OP is not outsourced. Let us now consider the situation when OP is hosted at XCorp site. In this case, you need to investigate architectures for access-in-place, data warehouses and migration at XCorp site and determine the appropriate Front-end Integration Components (FICs) and Back-end Integration (BICs) for each configuration. For example, data warehouses require extraction, transfer and load (ETL) of back-end data; access-in-place requires adapters/mediators, and migrations typically need a migration gateway. Additional rules are needed to suggest appropriate middleware technologies. For example, if the XCorp order processing application needs to access data from three inventory management systems then remote data access middleware can be used but for many systems an EAI platform is more appropriate.

2.4 Stage 4: Solution Evaluation

Solution evaluation goes into further details by translating the architecture A into plausible integrated solutions (S_1, S_2, \dots, S_n) with appropriate commercial-off-the-shelf (COTS) packages and performance/security/cost evaluations. The first step, COTS selection, allows the user to search the COTS database, part of the knowledgebase, and select the most appropriate solution based on cost constraints, the services needed and the technical interdependencies (for example, a .NET application does not work well in a Linux environment). Due to the complex interrelationships and interdependencies between different layers of technologies, the model shown in Figure 4 is used to analyze the process to process, process to app, app to app, app to middleware, and middleware to middleware interdependencies. Middleware to network and network to network integration is mainly considered in wireless integration projects where seamless services across Wi-Fi, cellular, and Bluetooth are needed [35]. We have specified several COTS selection rules that use this model and heavily rely on the current commercially available application and integration middleware (e.g., EAI software) and their interdependencies [22, 26, 32, 33, 34].

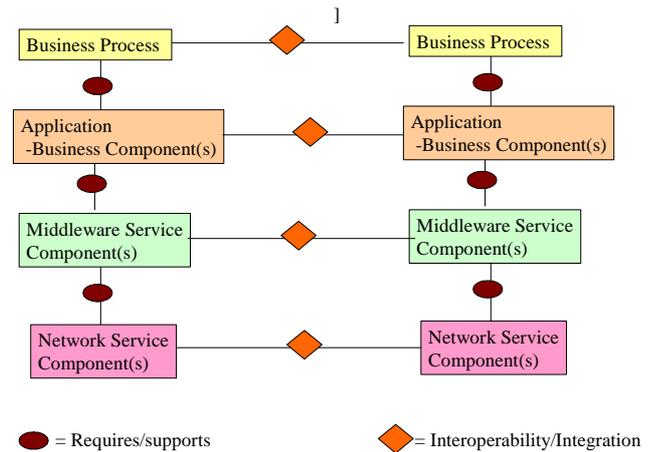


Figure 4: Interdependencies Model

In the next step, the solution S_i , as a result of COTS selection, is evaluated for performance and security issues. A performance analysis of FICs/BICS is conducted through an analytical model based on Little's formula [18]. For a security analysis of the FICs/BICS, the security issues due are investigated by using attack trees and security patterns [17, 36]. For more detailed security analysis, we are investigating the attack trees developed by the Amenaza's SecureITrees (www.amenaza.com). For cost estimation, the costs and benefits of all strategies are analyzed before reaching a final decision. The effort needed to integrate systems depends on the number and nature of integration components (FICs, BICs) identified by the IAA. From this, rough estimates of effort and cost can be obtained by using techniques similar to function point analysis.

3 PISA-AIM: The Decision Support Tool

The advisors of these modules, illustrated in Figure 1, collaborate with each other to develop IT plans and then analyze the architecture and integration aspects of the plan. Specifically, the PlanIT advisors do the following: the Enterprise Modeler develops a model of an enterprise, the Application Advisor develops an Application plan, the Platform Advisor develops a computing platform plan, the Network Advisor builds a network plan, and the Security Advisor develops a security plan. PlanIT is described elsewhere [38, 40]. The AIM advisors, described in this paper, help the user through life cycle of an integration project and develop an architecture and integration approach. PISA is supported through an extensive knowledgebase that contains a pattern repository, object models, and a COTS database. This knowledgebase also serves as a means of knowledge management by supporting queries and reports and consists of numerous strategies, patterns, COTS tools, project plans, and links to information sources. Common components (COTS Advisor, Project Planner, and Diagram Generator), support the overall system. Let us illustrate the operations of the AIM

modules through example of a manufacturing company (XCorp) introduced previously.

The **Business Problem Explorer (BPE)** helps the user to quickly build business process models and identify business scenarios that drive the specific integration project. After conducting high level analysis of different scenarios, the user proceeds by selecting BPs of interest for further exploration. At this stage, BPE heavily relies on the pattern repository (PR), part of the knowledgebase, to allow the users to review different aspects of the chosen BPs (e.g., process models, use cases, class diagrams, sequence diagrams, service descriptions, etc). These “knowledge chunks” serve as patterns that allow the users to quickly build more detailed and customized models by using tools such as UML.

The **Intelligent Requirements Generator (IRG)** conducts an interview, shown in **Figure 6**, which generates the company specific information for requirements document. The interview also captures the business scenario that drives the requirements. The outputs of this interview plus the knowledge chunks retrieved from the pattern repository are used to populate the various sections of the requirements document. The generated document is a requirements sketch in MS Word format that can be customized by the user.

The **Integrated Architecture Advisor (IAA)** suggests a technical architecture based on the requirements and walks the user through strategic decisions and scenarios of outsourcing, migrations, and data warehousing. A short interview helps the user analyze these strategies and suggest actions based on

the answers to questions such as the following: type of target application (queries, updates), types of resources needed from the back-end applications (data, processes), number of applications you need to integrate with (few, many), application type(s) you need to integrate with (legacy, new, mixture), and data currency requirements (low, medium, high). **Figure 7** shows a sample interview and a strategy recommended by IAA. Given an integration strategy, IAA also suggests a technical solution to support the strategy

The **Integrated Solution Advisor (ISA)** maps the technical architecture produced by IAA to COTS (commercial-off-the-shelf) solutions and evaluates the what-if scenarios for performance, security, and cost tradeoffs. The main results of these analysis show the estimated cost/effort, security and performance for different solutions (*S1, S2,..*) evaluated so far. **Figure 8** shows a partial output produced by ISA that shows performance information. The user can review this information and re-iterate to choose another solution if needed. At conclusion, ISA produces a detailed report that includes a project plan for the selected solution.

As stated previously, the AIM advisors are supported by an extensive knowledgebase of numerous strategies, hundreds of patterns, hundreds of COTS packages, dozens of customizable project plans, and the results of numerous user developed plans, solutions, and interviews stored as object models (OMs). The knowledgebase is organized in three parts: pattern repository, object models, and COTS database. Details of the knowledgebase are not given here due to space limitations.

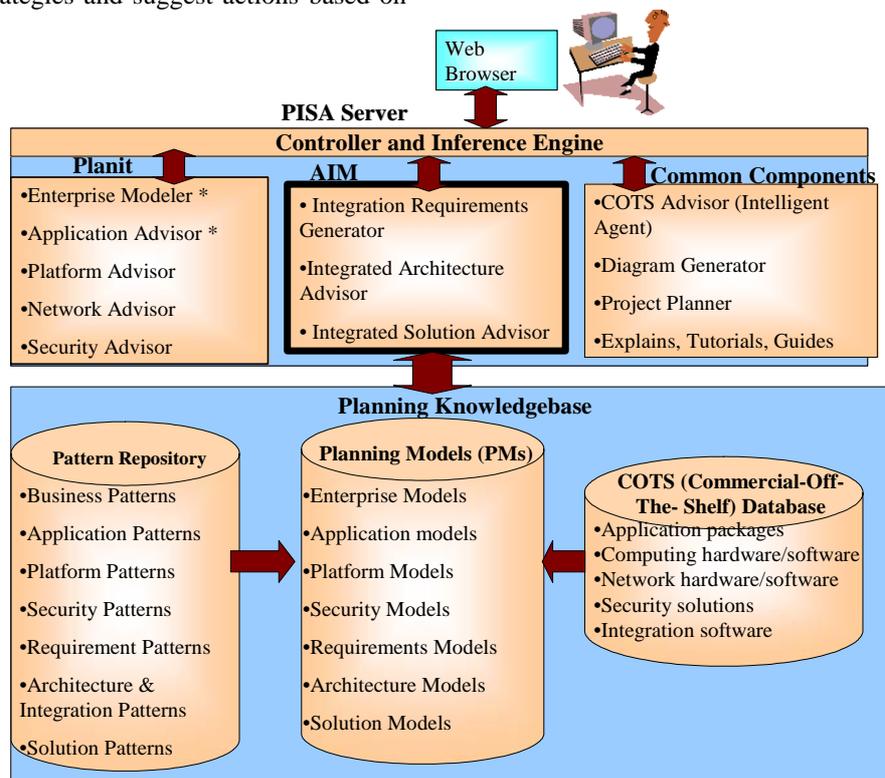


Figure 5: Conceptual View of PISA

Application chosen	Order Processing
User Access	<input type="radio"/> Web Access; <input type="radio"/> Wireless Handheld-Device Access; <input type="radio"/> Voice Access; <input type="radio"/> Application Access;
Number of back-end Applications Interfaced	None
Coupling with back-end Applications	Loose (Asynchronous)
Type of back-end Applications	Monolithic (legacy)
Services needed from back-end Applications	<input checked="" type="radio"/> Data <input type="radio"/> Function <input type="radio"/> Both
Data Translation for Back-end Application	No translation
Number of External Applications Interfaced for B2B	None
Interaction with External Applications	<input checked="" type="radio"/> Direct <input type="radio"/> Indirect (Intermediary)
Coupling with external Applications	Loose (Asynchronous)
Type of external Applications	Monolithic (legacy)
Services needed from External Application	<input checked="" type="radio"/> Data <input type="radio"/> Function <input type="radio"/> Both
Data Translation for External Application	No translation
Value of Transactions Handled by Application	None (\$0)
Volume of Transactions Handled by Application	Small (less than 10 per day)
Boundaries crossed in trade	<input checked="" type="radio"/> Local <input type="radio"/> National <input type="radio"/> International

Figure 6: Sample Interview for Requirement Generation

Application/BC Name	Order Processing
FACTORS TO BE CONSIDERED	CHOICES
Type of target application	Queries (Business Intelligence)
Type of resources needed from the back-end applications	<input checked="" type="radio"/> Data <input type="radio"/> Process <input type="radio"/> Both
Business value of back-end applications	<input checked="" type="radio"/> Low and Medium <input type="radio"/> High
Data currency requirements	Not strict (Daily or Weekly)
Flexibility needed in back-end systems	<input checked="" type="radio"/> Low <input type="radio"/> High
Number of back-end applications	<input checked="" type="radio"/> Few (less than 5) <input type="radio"/> Many (more than 5)
Type of back-end applications	Well structured (well defined interfaces)
Attitude towards external hosting	Not favored
<input type="button" value="Show Recommendations"/>	

Figure 7: Sample Interview for Strategic Analysis

Important Parameters				
Application:	Order Processing	Incoming Message size (assumed) =	100 bytes	
Local Datarate:	10 Mbps Ethernet	Outgoing Message size (assumed) =	500 bytes	
Internetwork (campus-campus) datarate:	T1	Local Server Processing rate =	20 queries / sec (assumed)	
Local Users:	10	Corporate Main Frame server processing rate =	100 queries / sec (assumed)	
Users at other Sites:	10	Utilization = Arrival Rate at server * Service time per message		
<input type="button" value="Calculate Performance Indicators"/>				
Performance Calculations				
Application	Backend Applications	Solution	Response Time (seconds per message)	Server Utilization (single application on server)
Order Processing		Allocated to local site server	0.0006	0.0060
		Allocated to Corporate MainFrame Server	0.0040	0.0460

Figure 8: Sample Output- Performance Evaluation from Integrated Solution Advisor

4 Comparison With Related Work and Concluding Remarks

A great deal of literature on enterprise architecture and integration projects (AIPs) has been published in the recent years under headings such as EAI (Enterprise Application Integration), MDA (Model Driven Architectures) and SOA (Service Oriented Architectures). This includes textbooks [19, 32], standards and consortia [42, 43], research papers [44, 45, 46, 47] and industry tools (e.g., IBM eBusiness Framework [15]). To locate an approach and/or a decision support tool that is somewhat comparable to PISA-AIM, we reviewed 11 textbooks, 83 research articles, 52 research and industrial tools, and numerous websites of industry consortia and standards bodies. The results were disappointing because we found no DSS tools for AIPs. Library and general Web searches with keywords like ‘DSS for architecture and integration’ yield articles that discuss how to integrate DSS *with* EAI (e.g., Lee [45]) but nothing about DSS *for* EAI. Specifically, a few models, under the heading of EAI have been proposed in the research literature to highlight different aspects of AIPs. For example, research challenges in EAI (e.g., how integration fits in ebusiness and ecommerce) are presented in [47], different EAI frameworks (e.g., Zachman, ISO Open Data Processing) are compared in [46], and a reference architecture for EAI is developed in [44] to improve understanding of the EAI concepts. However, these models basically illustrate the structural components of EAI systems and do not discuss how to develop a DSS for AIPs.

To conclude, this paper describes an intelligent decision support environment (PISA-AIM) that uses patterns, best practices, inferences, and collaboration for enterprise architecture and integration projects. A system of this nature has not been reported previously in the literature. The PISA-AIM system, operational as a beta version at present, has been used in four consulting assignments (others are in progress) and found it to be a very useful tool in exploring different business scenarios and evaluating the tradeoffs between integration strategies of outsourcing, data warehouses, and access-in-place. The IT managers especially appreciated the opportunity to develop some understanding of the options before embarking on an integration project. In addition, AIM has been used to teach six systems design, enterprise architecture and integration courses so far with very encouraging results. In each course, the students were assigned three projects: 1) manually develop an integrated architecture for an SMB that is going through a major re-engineering effort, 2) use AIM to solve the same problem, and 3) use AIM for a project of their own interest. Most students had a great deal of fun with the third project -- they built models of different businesses and developed integrated architectures by using AIM for “what-if” analysis of different scenarios. We are currently negotiating with

several universities and businesses for additional experiments.

We are using our current experiences and lessons learned to guide future research and development directions. To be realistic, we are following an iterative approach where each iteration adds more depth and intelligence to the advisors, expands the current knowledgebase through improved inference and learning (e.g., case-based reasoning) techniques, and increases automation by generation of sample code where possible (e.g., WSDL for service definitions and front-end/back-end adapters for integration). We intend to extend the business process model to capture more business intelligence based on developments in business ontologies and business process definition languages and environments such as ARIS, BPML, WS-BPEL, XPDL, and UML 2.0 [7, 39]. We are also accumulating more patterns, refining the existing ones, and using them in more situations. An interesting research area is to automatically generate “optimal” solutions by taking advantage of the extensive knowledgebase. For example, a solution could be generated that minimizes cost, performance delays, or certain security threats based on some constraints. Naturally, we want to expand our current focus from SMBs to large businesses.

Acknowledgement

Members of the NGE Solutions team (Dolores Umar, Kamran Khalid, Adnan Javed, Adalberto Zordan, and Nauman Javed) have significantly contributed to the development of this project by building and testing the PISA-AIM system..

References

- [1] Adams, J., et al, "Patterns for e-Business: A Strategy for Reuse", IBM Press, October 2001.
- [2] Alexander, C., "The Timeless Way of Building", Oxford University Press, 1979
- [3] Alexander, C. et al , "A Pattern Language", Oxford University Press, 1977
- [4] Brodie, M. and Stonebraker, M., "Migrating Legacy Systems", Morgan Kaufman, 1995
- [5] Boehm, B., and Abts, C., "COTS Integration: Plug and Pray". Computer, vol. 32, no. 1, 1999, pp. 135-138.
- [6] Buschmann, E., et al, "Pattern-Oriented Software Architecture, Vol. 1: A System of Patterns", John Wiley, 1996.
- [7] Business Process Definition Languages -- An Analysis (<http://www.ebpml.org/status.htm>)
- [8] Ericksson, H. and Penker, M., "Business Modeling with UML – Business Patterns at Work", John Wiley, 2000.
- [9] Ferdinandi, P., "A Requirements Pattern: Succeeding in the Internet Economy", Addison-Wesley, January 2002.
- [10] Fox, M. and Gruninger, M., "On Ontologies and Enterprise Modeling", International Conference on Enterprise Integration Modeling, 1997
- [11] Gamma, E., et al, "Design Patterns", Addison Wesley, 1994.
- [12] Hagge, L. and Lappe, K., "Sharing Requirements Engineering Experience Using Patterns", IEEE Software, January-February 2005, pp. 24-31.
- [13] Hohpe, G. and Woolf, B., "Enterprise Integration Patterns : Designing, Building, and Deploying Messaging Solutions", Addison-Wesley, 2003.
- [14] Hull, E., "Requirements Engineering", Springer 2004
- [15] IBM e-Business Framework website <http://www-106.ibm.com/developerworks/patterns/>
- [16] Kalakotta and Robinson, "E-Business 2.0", Wiley, 2002
- [17] Kienzle, D., and Elder, M., "Security Patterns for Web Development", DARPA Contract No: F30602-01-C-0164, June 2001. Weblink: <http://www.scrpt.net/~celer/securitypatterns/final%20report.pdf>.
- [18] Kleinrock, L., "Queuing Systems - Vol. 2", John Wiley, 1976.
- [19] Linticum, D., "Enterprise Application Integration", Addison-Wesley Information Technology Series, 1999
- [20] Maiden, N.A., and Neube, C., "Acquiring COTS Software Selection Requirements", IEEE Software, vol. 15, No. 2, 1998, pp. 46-56.
- [21] Maedche, A., et al, "Ontologies for Enterprise Knowledge Management", IEEE Intelligent Systems, 2003
- [22] Missier, P. and Umar, A., "Representing Knowledge about Modern Software Architectures", IFIP international conference on Knowledge-based systems (April 2001).
- [23] Patterns Web Site, <http://hillside.net/patterns>
- [24] Ovum Report, "Enterprise Application Integrators", Ovum Group, 1999.
- [25] Schneier, B., "Attack Trees", Dr. Dobb's Journal, Dec., 1999.
- [26] Schmidt, D., "Real Time Embedded Systems Research", Institute for Software Integrated Systems, Vanderbilt University, www.isis.vanderbilt.edu, Presentation: August 2004.
- [27] Sommerville, I., "Integrated Requirements Engineering", IEEE Software Magazine, Jan-Feb. 2005, pp. 16-23
- [28] Sneed, H., "Planning the Reengineering of Legacy Systems", IEEE Software, January 1995, pp. 24-34.
- [29] Umar, A., and Missier, P., "A Knowledge-based Decision Support System for Enterprise Integration", First International Workshop on Enterprise Management & Resource Planning Systems, Venice, Nov. 1999.
- [30] Umar, A., and Missier, P., "A Framework for Analyzing Virtual Enterprises", Research Issues in Data Engineering (RIDE) Workshop on Virtual Enterprises, March 1999.

- [31] Umar, A., "IT Infrastructure to Enable Next Generation Enterprises", *ISF (Information Systems Frontiers) Journal*, Volume 7, Number 3, July 2005, pp: 217 - 256.
- [32] Umar, A., "e-Business and Distributed Systems Handbook: Integration Module", NGE Solutions, May 2003, revised August 2004.
- [33] Umar, A., "e-Business and Distributed Systems Handbook: Architecture Module", NGE Solutions, May 2003, revised August 2004.
- [34] Umar, A., "e-Business and Distributed Systems Handbook: Middleware Module", NGE Solutions, May 2003, revised August 2004.
- [35] Umar, A., "Mobile Computing and Wireless Communications", NGE Solutions, July 2004.
- [36] Umar, A., "Information Security and Auditing in the Digital Age", NGE Solutions, May 2003, Revised August 2004.
- [37] Umar, A., "Optimal Program and Data Allocations in Distributed Environments", Ph.D. Dissertation, Univ. of Michigan, 1984
- [38] Umar, A., et al, "Computer Aided Consulting for SMBs", IRMA 2005 Conference, May 2005
- [39] Vemadat, F., "Enterprise Modeling: Objectives, Constructs, and Ontologies", Tutorial posted on INTEROP site (<http://www.interop-noe.org/>).
- [40] Web Link1: PISA Website www.ngesolutions.com/pisa
- [41] Web Link2: ISO Software Requirements Template, <http://www.12207.com/templates.htm>
- [42] Web Link3: MDA site (www.omg.org/mda)
- [43] Weblink4: IDEAS (Interoperability Development for Enterprise Application and Software) Roadmap: "State of the Art Summary", www.ideas-roadmap.net, deliverable D1.1, Mat 2003.
- [44] Chalmeta, R., Campos, C. and Grangel, R., "References architectures for enterprise integration", *Journal of Systems and Software*, Volume 57, Issue 3, 15 July 2001, Pages 175-191.
- [45] Lee, J., Siau, K., and Hong, S., "Enterprise Integration with ERP and EAI", *CACM*, 2003, vol. 46, no. 2, pp. 54-60.
- [46] Losavio, F., Ortega, D., and Pérez, M., "Comparison of EAI Frameworks", *Journal of Object Technology*, Vol. 4, No. 4, May-June 2005, pp.93-114.
- [47] Sharif, A., et al, "Integrating the IS with the enterprise: key EAI research challenges", *The Journal of Enterprise Information Management*, Volume 17 Number 2 2004 pp. 164-170

Appendix A: Formal AIP Model

Figure 9 shows a formal view of the architecture and integration project (AIP) model shown in Figure 1. The main inputs of this decision model are:

- **Enterprise parameters** $E = \{e_t, e_s, e_d, e_w, e_m, e_a\}$ where e_t = company type (e.g., manufacturing), e_s = company size in terms of number of employees,

e_d = company distribution (e.g., local, regional, international), e_w = reliance on the web to conduct business, e_m = reliance on mobility to conduct business, and e_a = reliance on agility (e.g., on-demand services) to conduct business. These core parameters, as shown in Table 1, impact all stages of the AIP model. Thus by specifying or changing these few parameters, the planner can quickly investigate a new business scenario by generating application plans, requirements documents, and architectural configurations.

- **Patterns**, introduced by Alexander [2, 3], play an important role in this process. We specifically use business process patterns (BPPs) [1, 16], application and requirement patterns (APs) [9, 12], architecture patterns (HPs) [6], and integration patterns (IPs) [13] to provide generic sketches that are customized to produce a company scenario specific solution. The planning knowledgebase, shown and discussed in Section 3, contains these patterns and other models needed to support the decision model. Customization of patterns is based on the enterprise parameters E , stage specific inputs and PM s accumulated in the planning knowledgebase.
- **Local input parameters** W, X, Y and Z that are provided by the user in the four stages of the AIP model (see Figure 9). These parameters are used to provide the stage specific information, if needed. AIM automatically provides a set of reasonable defaults for these parameters based on patterns. Thus, if the user is in a hurry, a default solution sketch can be created extremely quickly (within 10 minutes). However, the user may override the default parameters for more customized analysis.

The main output produced by this process is the **planning model** PM , a set which consists of several subsets where each subset represents results of an AIP stage. In the beginning, PM is a simple sketch that is successively enriched as the user progresses through different stages of the proposed AIP model. At the conclusion of an interview, a complete company specific plan is represented in the PM , i.e., $PM = [M, R, A, S]$ where M, R, A , and S represent the enterprise model, the application requirements plan, the integrated architecture configuration, and the integrated solution, respectively.

Given the inputs E that represent enterprise parameters, (W, X, Y, Z) that represent stage specific information and the patterns (BPP, AP, HP, IP), the following relationships are used to create a planning model $PM = [M, R, A, S]$ that represents the complete AIP:

$$\begin{aligned} M &= f(E, W, BPP) \\ R &= f(E, M, X, AP) \\ A &= f(E, R, Y, HP) \\ S &= f(E, A, Z, IP) \end{aligned}$$

The planning model PM , as shown in Figure 9, is constructed successively in various stages of the model. Note that the E impacts all outputs. Table 1 displays details of the interrelationships between E

and the stages of the AIP model and presents the theoretical foundation of the proposed AIP model. Each cell of this table has been converted into a set of rules. Specifically, each function f , e.g., $f(E, W, BPP)$, represents a set of rules that are discussed qualitatively in the main body of the paper. These rules represent the core inference net used by AIM.

Note that E is the *only* input required from the user, all others use defaults and/or patterns from the knowledgebase. Thus the planner can adjust a few parameters in E (e.g., introduce on-demand services, increase number of sites, etc) for new business scenarios and quickly run through all stages of an AIP.

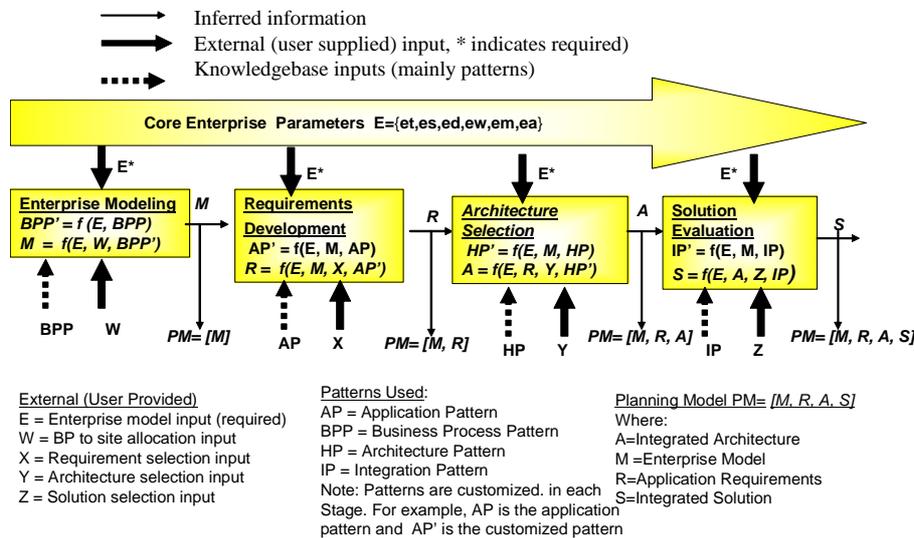


Figure 9: Formal View of the Architecture and Integration Project (AIP) Decision Model

Table 1: Core Parameters and their Impact on Integration Planning

Core Parameters $E = \{e_t, e_s, e_d, e_w, e_m, e_a\}$	Enterprise Model M Produced	Application Requirements Plan R Produced	Integrated Architecture plan A Produced	Integrated Solution Plan S Produced
e_t = Company Type (e.g., manufacturing)	$e_t \rightarrow BPP \rightarrow M$ e_t determines business process pattern (BPP) which is the foundation of enterprise model M	$e_t \rightarrow BPP \rightarrow AP \rightarrow R$ e_t and BPP determine the application patterns (APs) that specify the type of application packages needed (e.g., payment package for payment). AP determines R	$e_t \rightarrow BPP \rightarrow R \rightarrow A$ e_t and BPP influences the type of interfaces (batch, interactive) between BPs and determines requirements model R. R determines A.	$e_t \rightarrow A \rightarrow S$ e_t impacts security exposure (e.g., financial BPs need higher security) but does not have direct impact on cost and performance.
e_s = Company Size: Number of Employees (Low, Medium, High)	$e_s \rightarrow BPP \rightarrow M$ e_s customizes business process pattern (BPP) because larger companies may have additional BPs. BPP is used to build M	$e_s \rightarrow AP \rightarrow R$ e_s influences application pattern AP (e.g., small companies need a few BPs to be automated through applications). AP drives R.	$e_s \rightarrow R \rightarrow A$ e_s influences requirements model R (e.g., integrated architectures for large scale systems use expensive EAI platforms).	$e_s \rightarrow S$ e_s impacts security (larger companies have higher impact) and cost (more users need expensive solutions).
e_d = Company Distribution: Local, Regional, International	$e_d \rightarrow BPP \rightarrow M$ e_d customizes business process pattern (BPP) because highly distributed companies may have additional BPs for international trade. BPP is used to build M	$e_d \rightarrow AP \rightarrow R$ e_d determines the application pattern AP (e.g., transaction processing and workflow packages are needed for compliance for international laws, taxes and standards, e.g., I18N).	$e_d \rightarrow R \rightarrow A$ e_d helps determine the requirements model R to reflect sophistication of applications needed to support B2B trade (robust supply chain management for many partners).	$e_d \rightarrow S$ e_d impacts security exposure (B2B, international) and could impact cost and performance (widely distributed systems can be more expensive)
e_w = Reliance on Web (Low, Medium, High)	$e_w \rightarrow BPP \rightarrow M$ High value of e_w customizes BPP to include more eBusiness BPs and BPP is used to build M	$e_w \rightarrow AP \rightarrow R$ High value of e_w indicates strong eBusiness and possibly real-time business activity monitoring (RBAM) support (this influence AP)	$e_w \rightarrow R \rightarrow A$ High value of e_w requires high degrees of Web integration (this is reflected in requirements model R). R determines A.	$e_w \rightarrow S$ High value of e_w impacts security exposure, cost and performance of a solution.
e_m = Reliance on Mobility (Low, Medium, High)	$e_m \rightarrow BPP \rightarrow M$ High value of e_m introduces Mobile Business BPs in BPP and BPP is used to build	$e_m \rightarrow AP \rightarrow R$ High value of e_m requires M-Business applications (e.g., M-CRM, M-SCM, M-Portal)	$e_m \rightarrow R \rightarrow A$ High value of e_m requires front-end integration (this is reflected in	$e_m \rightarrow S$ High value of e_m requires stronger security, increases cost

	enterprise model M	for high mobility and location-based services.	requirements model R). R determines A.	and raises performance issues due to wireless. .
e_a =Reliance on Agility (e.g., On-demand services)	$e_a \rightarrow BPP \rightarrow M$ High value of e_a introduces new BPs for agility in BPP and BPP determines M	$e_a \rightarrow AP \rightarrow R$ High value of e_a requires component-based apps for flexibility(reflected in AP).	$e_a \rightarrow R \rightarrow A$ High value of e_a requires SOA (reflected in R). R determines A.	$e_a \rightarrow BPP \rightarrow S$ High value of e_a increases security exposure