

Enterprise Architecture Patterns for Multichannel Management

Marc M. Lankhorst, Paul H.W.M. Oude Luttighuis

Telematica Instituut
PO Box 589
7500 AN Enschede
The Netherlands

{marc.lankhorst, paul.oudeluttighuis}@telin.nl

Abstract: This paper describes a catalogue of patterns for multichannel management: functional structures for designing organizational and technical solutions that help organizations to manage and align the various information channels they use in communicating with their customers. The paper outlines the structure of the catalogue and the identification of the patterns, and describes an example pattern in more detail to give the reader an impression of the catalogue's content.

1 Introduction

Since the early nineties, organizations have been using a variety of customer service channels. Next to the traditional channels, such as mail, fax, reception desk or telephone, customers have access to digital channels like websites and e-mail. These service channels have different characteristics and are used for communication, interaction, transaction and distribution of products and/or services. Channel usage statistics suggest customers still prefer more expensive personal channels (desk and telephone) over newer cost-effective digital channels.

Many organizations, especially in government, struggle with the integration and management of these service channels. In particular, channel synchronization needs to be addressed. Channel synchronization and coordination is required as customers expect information and services to be consistent across channels. Channel synchronization is not only a technical but also an organizational problem, since information needs to be shared, processes must be aligned, and the customer should be addressed in a similar fashion across various parts of the organization. Solutions to these problems are usually developed on an ad hoc basis, without a solid methodological foundation. In order to deploy customer service channels effectively and efficiently, architects of multichannel management solutions need to be supported in a better way. So far, there are very few sources of proven enterprise architecture knowledge about multi-channel management.

In software engineering and architecture, patterns have become a popular way of capturing architectural knowledge [Al77, Fo02]. The notions of design patterns and architec-

ture patterns, however, are hardly applied to enterprise architecture, of which we consider multi-channel architecture to be a constituent. Still, the architectural complexity of multi-channel management, outlined in this article, provides sufficient basis for the hypothesis that patterns might help to manage this complexity. This paper reports our efforts of developing a catalogue of architecture patterns for multichannel management.

To clarify what we mean by multichannel management, we first address the aspects that are relevant in any organization using multiple channels for its communication with customers. Based on these, we then define a framework of design issues for which architecture patterns may be useful. We explain one of these patterns in more detail to illustrate our approach and report on the practical application of these patterns.

2 Characteristics of Multichannel Management

In multichannel management, and hence in multichannel architecture, we observe four core elements that need to be addressed: *clients*, *services*, *providers*, and *channels*. These four key dimensions extend the three basic concepts of service architectures (service, client and provider) with the central concept in multichannel management (channel).

In multichannel situations, different *client* groups may have different needs, demands and preferences with respect to e.g. service content, channel usage, and parties involved. Also, different *service* characteristics may influence channel usage:

- Group information services: these offer generic information to groups of (usually anonymous) customers. Think of e.g. marketing campaigns via radio or TV, generic mail to customers, providing information via the Web, et cetera.
- Individual information services: these provide specific (personalized) information that is targeted to an individual, identifiable customer. Think of e.g. the progress of the client's application for a residence permit, events taking place in his neighbourhood, et cetera.
- Transaction services: these provide a specific product or service to an individual, identifiable customer and require some kind of contractual relation between service provider and customer. Think of e.g. buying a book from a Web shop, applying for child benefits, et cetera.

Each service is offered by one or more *service providers*. Often several parties are involved. We therefore distinguish between three responsibilities in the service delivery process: accountability, coordination, and execution. Especially in complex service delivery networks such as those in government, these responsibilities may be distributed across different organizations. Take for instance a typical re-integration scenario, designed to aid the unemployed in finding their way back to the labour market. In this case, the Ministry of Labour may be accountable, a social security agency may coordinate, and a private re-integration agency may provide the actual service of coaching people back to a job. All these parties must comply with legal restrictions and other regulations, for example on privacy protection, archiving laws, or bookkeeping standards.

Finally, each service is provided through one or more *channels*, such as mail, telephone, website, e-mail, sms, chat, front desk, mass media, intermediaries, and more. Of course,

these channels differ in many respects, such as opportunities for personalization, immediate feedback or not, social influence, geographical and social reach, amount of information that can be delivered, cost per contact, and synchronicity of communication (see also [Te07]). Choosing the ‘right’ channel(s) for a specific service and target group, however, is beyond the scope of this paper.

These four dimensions of client, service, provider, and channel span the multi-channel architecture problem space. As explained in [Fi08], a large part of the complexity of multichannel management is in the multiplicity of these relations, i.e., situations in which multiple clients, services, providers, and/or channels are involved (the latter being ‘multichannel management’ in the strict sense). Take again our reintegration example, in which several government institutions, both local and national, together with private companies like temping agencies and job banks, communicate with the customer through telephone, mail, e-mail and website. All these parties and channels involved must be synchronized to provide the client with a seamless experience.

A number of important forces are at work in this problem space. First, there are the clients’ needs for *unified service provision* across multiple channels and organizations. This is counteracted by the *organizational subdivision* of government and other organizations along different lines, often related to legal tasks and responsibilities or to areas of expertise, and not to the clients’ needs. Another important force is that of the *dissimilar channel and service characteristics*. For example, whether a channel is synchronous or asynchronous influences the way in which applications that support these channels deal with process coordination and data storage. Finally, there is the force (or deadweight...) of *legacy systems*, which often requires sub-optimal solutions from an idealistic point of view. The patterns we have identified aim to balance these forces.

3 Framework for Identifying Multichannel Management Patterns

To identify potentially useful patterns for multichannel management, we did not want to go about in an ad-hoc manner. We have therefore defined a framework that positions the various aspects of multichannel architectures for which patterns could be relevant. The first dimension of this framework is given by the four aspects distinguished in Section 2, refined further by using a number of interrogatives, analogous to e.g. the Zachman Framework [Za87]:

- *Who* is the client and what do we know about him/her?
- *What* service(s) does the client consume?
- *Where* are these services provided, i.e., through which channels?
- *Whose* services are these, i.e., who provides them?
- *With whom* are these services provided or consumed, i.e., who provides assistance?
- *How* are these services delivered and realized?

A second important dimension is that of the elements of the service delivery process. During this process, the (prospective) client first searches and *selects* the right service, channel and/or provider. This might happen for example via a website offering these services, via an intermediary assisting a client, or via an interactive voice response sys-

tem. Next, the chosen service is *delivered* to the client via the chosen channel of the service provider. To this end, the provider must perform all kinds of internal processes and functions, using various automated systems in the ‘back-end’, to actually *realize* this service. This service delivery process is not a perfect funnel: during service delivery, for example, responsibility may pass from one provider to another, and in every step different channels may be used.

Based on these aspects of multichannel management on the one hand and the service delivery process on the other hand, we have created a framework for identifying multichannel management patterns. Each of the cells in the table represents a relevant area of design, and hence a possibility for discovering one or more architecture patterns. Later in this paper, we will show how this framework is filled in with relevant patterns.

4 Structure of Multichannel Solutions

Architecture patterns abstract from specific technological solutions, but focus on functionality needed to solve a given architectural problem. In order to organize the structure of multichannel management solutions, and hence of the architecture patterns we have identified, we have employed a layered framework, in which specific functional aspects of multichannel service provisioning are positioned:

- The (objective) *situation* of the client. What are his needs, personal situation, profile, circumstances? But also: what communication channels can he use?
- The *preferences* of the client. What demands and desires does he have with respect to service content, manner of delivery, channels used, et cetera?
- The *dialogue* with the client. Starting from the client’s situation and preferences, his actual needs and the applicable (combination of) service(s) must be established.
- The *channel(s)* used to provide information and services. This includes the medium used (e.g. paper, telephone, Web, face-to-face), but also differences within these categories, like using large fonts for communicating with visually impaired clients, a language switch on a website used by non native speakers, et cetera.
- The *content* of the information and services. Often this consists of textual descriptions like brochures, forms, web pages and call centre scripts, if necessary targeted to specific client groups.
- The *business logic* of the information and services. This comprises knowledge models, concepts, procedures, computations and other business logic that determines the rights and obligations of clients, the form in which a service or product is delivered, the steps needed in service provisioning, checks to be made, deadlines to be met, et cetera.
- The *data* that is used, comprising client data, case data, product data, et cetera.
- The *communication* within and between the organizations involved.

As said, these layers describe *functional* aspects, i.e., the functions that must be carried out to provide a service. Hence, the service itself is not a separate layer in this framework, but rather an abstraction of these functions (or vice versa, these functions jointly realize the service). The layered model serves as a structural template and check list for

the multichannel architecture patterns. Next to these functional aspects, we have also identified aspects that address management, control, and realization of multichannel solutions. However, these aspects are deemed out of scope for this paper. For more details, the reader is referred to [La08]. The layers in this framework help us in describing the content of the multichannel patterns and position them relative to each other. Some patterns are more ‘technical’ or ‘internal’ in nature, covering the lower layers of the framework, others are more ‘customer-oriented’, focussing on the upper layers.

The patterns we have identified are at a somewhat more detailed level of abstraction than for example IBM’s e-business patterns [Ko01], which contain high-level structures such as ‘self-service’ or ‘extended enterprise’ and some more detailed patterns within these. There are similarities between the two approaches, in particular in IBM’s ‘access integration’ pattern, but its subordinate patterns are strictly application-oriented, and their work does not focus on general multichannel solutions.

5 Catalogue of Multichannel Management Patterns

In the previous sections, we have outlined relevant characteristics of multichannel architectures and the resulting design issues and aspects for which architecture patterns may be of use. In this section, we describe the contents of our pattern catalogue.

5.1 Description of Patterns

The description of the multichannel patterns in the catalogue contains the following elements, inspired of course by [Ga95], [Bu99] and many other pattern catalogues:

- **Name:** A short descriptive name of the pattern.
- **Context:** In which circumstances is this pattern useful?
- **Problem:** Which problem does this pattern solve?
- **Solution:** What solution does this pattern offer?
- **Forces:** Which forces have to be balanced?
- **Structure:** What is the architectural structure of the pattern?
- **Consequences:** How does the pattern resolve the forces?
- **Known uses:** How is this pattern used in practice?
- **Related patterns:** How is this pattern linked to other patterns?

The pattern’s structural description typically includes business roles, business processes, services, application functions, data collections and other functional elements. The pattern’s structure is also mapped onto the layers of Sect. 4.

5.2 Pattern Identification

To identify relevant architecture patterns for multichannel management, we started from the framework of relevant multichannel architecture issues described in Sect. 3. To find patterns that fit these issues, we have looked at applicable literature (e.g. [Bu99], [Fo02],

[HW04], [Mc06], [Bo08] and more), investigated real-life architectures, especially in different governmental organizations in the Netherlands (e.g. [Ba07], [Eg06]), and examined various existing technological solutions in which useful patterns are apparent [He08]. This investigation has led us to the identification of some 30 patterns, which are shown in Table 1. A number of these patterns spread across multiple architecture issues from our framework. For example, the PERSONALIZATION pattern provides a rather generic solution for dealing with the use of personal profile data, and hence covers several cells of the framework.

Table 1. Identification of multichannel management patterns.

ASPECT	SELECTION		DELIVERY	REALIZATION
Who?			ACCESS	CENTRAL ADMINISTRATION VIRTUAL DOSSIER BUSINESS INTELLIGENCE
			DELEGATION	
What?	SERVICE SELECTION		PERSONALIZATION	CASE MANAGEMENT DOCUMENT MANAGEMENT CONTENT MANAGEMENT
				SERVICE COMBINATION
Where?			CHANNEL COMBINATION	CHANNEL STACK
Whose?	FUNNEL	REDIRECTION	CASE TRANSFER	
	SERVICE TEAM			
With whom?	INTERMEDIARY		CO-BROWSING	PUBLISH-SUBSCRIBE
How?	PORTAL		WIZARD	KNOWLEDGE BASE RULE ENGINE BUSINESS INTELLIGENCE MID-OFFICE BUSINESS PROCESS MGT. OPERATIONAL DATA STORE
			ELECTRONIC FORM	

These patterns are explained in more detail in Table 2. Many of these are concerned with the synchronization of content, channels, and/or providers, to ensure a seamless experience for the client. As we can observe, these patterns have different granularities. Some patterns address localized, small problems, others higher-level architectural decisions. Like the pattern language suggested by Alexander [A177], our catalogue of interrelated patterns helps architects to solve design problems at different levels of abstraction. An example is the MID-OFFICE pattern, which links front- and back-office systems and processes. This uses other, smaller-scale patterns such as DOCUMENT MANAGEMENT and BUSINESS PROCESS MANAGEMENT that address specific aspects within the general problem field of front- and back-office integration.

Table 2. Description of multichannel management patterns.

NAME	DESCRIPTION
SERVICE SELECTION	How to help clients select the right services for their needs? Ask the client relevant information and offer him explicit choices of service, provider and/or channel
PERSONALIZATION	How to adapt the selection, delivery and realization of services to the client's situation, needs and desires? Maintain a personal client profile with relevant personal data.
CHANNEL COMBINATION	How to use different channels that offer complementary information or interaction? Use two or more channels synchronously, e.g. calling up a company while browsing through its website
CHANNEL STACK	How to let a client use a channel without having access to that channel himself? Use one channel through another, e.g. a call centre agent who uses the website on behalf of the client
CO-BROWSING	How to assist a client in using the website? Let an employee co-browse, i.e., visit the website together with the client
SERVICE COMBINATION	How to combine services of different organizations through the same channel? Use a separate process to offer a combined service
INTERMEDIARY	How to advise and assist clients with choosing an using services from different providers? Use an intermediary as a third party
SERVICE TEAM	How to offer integrated services across channels? Use a service team that combines front- and back-office activities
FUNNEL	How to efficiently use scarce expert knowledge in handling client questions? Funnel these requests depending on the complexity of the question, via the website, call centre, to back-office specialists
REDIRECTION	How to help a client find the right channel and/or provider no matter where he asks his question? Redirect him directly to the applicable channel/provider
CASE TRANSFER	How to hand over a client case to a provider that offers another part of the service? Transfer the case responsibility and case data
ACCESS	How to provide secure access to data and services? Use client identification, authentication, and authorization
DELEGATION	How to have someone else represent a client? Have the client delegate the authorization to use a service to someone else, e.g. to an intermediary or a family member
WEB PORTAL	How to offer services from different providers in a single website? Use a web portal
ELECTRONIC FORM	How can a client submit relevant information online? Use an infrastructure for structured electronic forms
WIZARD	How to assist a client online with using services? Provide an automated assistant that helps with finding the right service, filling in forms, et cetera
KNOWLEDGE BASE	How to storing and maintain business logic and other knowledge behind different products and services and facilitate its reuse? Use a central knowledge base
RULE ENGINE	How to encode and execute business logic, while ensuring flexibility and adaptability? Use a rule engine and encode the logic in formalized business rules
MID-OFFICE	How to link application and process silos in a fragmented back-office to integrated services provided through multiple channels in

	a unified front office? Use a mid-office layer for process and data integration
BUSINESS PROCESS MANAGEMENT	How to control the execution of (partially) automated business processes, separate from the application logic? Use a business process management solution, encoding processes in executable form
CONTENT MANAGEMENT	How to ensure consistent content across channels and services? Use content management to maintain content in a central place
DOCUMENT MANAGEMENT	How to provide employees with access to all correspondence with clients? Use central, digitized storage and access of all incoming and outgoing documents
CASE MANAGEMENT	How to provide employees access to all information on ongoing client cases? Use a centralized case management solution, accessible by all employees handling this case
VIRTUAL DOSSIER	How to bring client data together from different sources? Use indices and synchronization (instead of centralized storage) creating a virtual dossier
CENTRAL ADMINISTRATION	How to ensure consistent client data, for usage across different channels, applications, departments and organizations? Use a centralized client data administration
OPERATIONAL DATA STORE	How to provide 24x7, secure access to data in non-24x7 back-office systems? Use an operational data store that caches caches or copies back-office data for use by front-office applications
BUSINESS INTELLIGENCE	How to obtain management data regarding channel and service usage? Measure and analyse current and past client data to provide information to support management decisions
PUBLISH-SUBSCRIBE	How to distribute changes in data automatically to keep client and service data in sync across channels? Let systems register their interest and publish relevant changes to them

The catalogue also describes the relations between the various patterns in our pattern language. It contains three main types of relations, next to the subdivision into the six interrogative categories:

1. Patterns that often *occur together*, such as PORTAL and ELECTRONIC FORM (such forms being used in Web portals), CONTENT MANAGEMENT and DOCUMENT MANAGEMENT (different forms of digital asset management, which may e.g. share (parts of) their implementation), or RULE ENGINE being used with PUBLISH-SUBSCRIBE, for event filtering in an event-based architecture. These relations are shown with undirected arcs.
2. Patterns that provide *potential realizations* of (parts of) other patterns at a higher abstraction level. Examples include MID-OFFICE and its sub-patterns described in the next section, INTERMEDIARY as a way of realizing SERVICE SELECTION, CASE TRANSFER using VIRTUAL DOSSIER or CENTRAL ADMINISTRATION to share client data being transferred, or PUBLISH-SUBSCRIBE as a way of realizing the communication within a VIRTUAL DOSSIER. These sub-patterns are shown in a smaller font.
3. Patterns that provide *alternative solutions* to a similar problem, such as VIRTUAL DOSSIER and CENTRAL ADMINISTRATION, or SERVICE TEAM and FUNNEL. These relations are shown with double-headed arcs.

6 Example: MID-OFFICE and its Sub-Patterns

In this section, we give a (somewhat abbreviated) description of the MID-OFFICE pattern, according to the pattern template outlined in Section 5.1.

Context: In many organizations, the back office is organized along the lines of the applicable expertise, both in an organizational sense and in terms of the application landscape. This does not fit the needs of clients, since these needs may span multiple expertise areas, systems and organizational units. To address these client needs in an integral way, a unified front office is often created, comprising e.g. the call centre, website, front desk et cetera. This must then be linked to the existing, fragmented back office.

Problem: How do you link application and process silos in a fragmented back-office to integrated services provided through multiple channels in a unified front office?

Solution: Use a MID-OFFICE, an intermediate layer of functionality that handles both process and data integration between the front office and the back office.

Forces: The two main forces at play in this pattern are the client's need for *unified service provision* versus the *organizational* (and consequently the application landscape's) *subdivision* along lines of expertise. An integrated view of the client situation is needed to provide adequate products and services. Treating a client request, however, may require many different areas of expertise, each often using dedicated systems.

A second important force is the need for *integrating legacy applications* from the back office. Although a fully service-oriented architecture might be the 'ideal' solution in the long term, until then a short-term solution for the integration of these systems must be found. Point-to-point links between multiple channels and multiple systems are too costly to maintain and do not scale well to more systems and/or channels. Hence, a mid-office architecture is often used as a temporary phase in the migration from monolithic systems to a SOA.

Other relevant forces opposing the need for legacy integration are *availability*, *security* and *performance*: back-office systems are often not available around the clock, are not meant to be accessed directly by clients for security reasons, and are not designed to handle the load of large numbers of clients. Directly linking these systems to e.g. the website is therefore not an option.

Structure: Figure 1 shows the main structure of a MID-OFFICE. This and the other patterns in the catalogue are described using the ArchiMate modelling language [La05] for enterprise architecture; the arrows in this figure denote usage of services, the boxes denote applications. The pattern contains four core elements:

- A *case warehouse* that stores the data on ongoing cases of clients, where a case may span multiple expertise areas and systems. Thus, it provides front-office systems and employees with a unified view on the client's situation and needs. It offers services for case file access to the front-office applications (e.g. Web portal, call centre).
- An *operational data store* that ensures consistency of data from multiple sources. It provides a unified, secure and highly available cache to the client, object and prod-

uct data in back-office applications and offers services to the case warehouse and front-office applications to access these data.

- A *business process engine* that coordinates the service provision process across multiple systems and departments, using a case-based approach. This ensures a coordinated response to the client's needs by all systems and employees involved. Individual back-office departments may of course also have their own business process systems for managing their internal processes.
- A *document management system* that holds all incoming and outgoing correspondence, which can be accessed by the case warehouse. This facilitates a fully digitized workflow needed to have an integral view on the client's situation.

The back-office applications offer services to the mid-office applications to get access to client, object and product data, and if necessary they use the correspondence stored by the document management system.

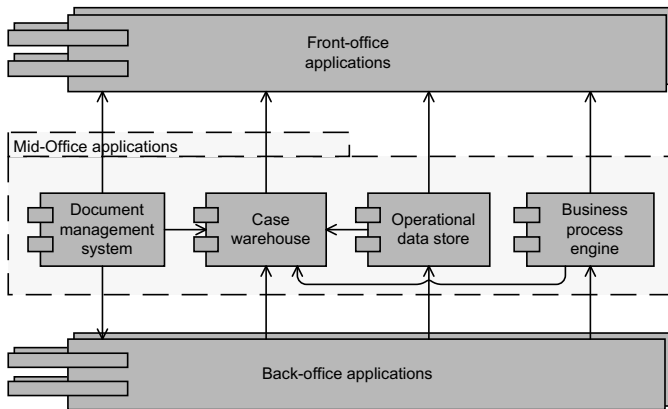


Figure 1. Basic structure of the Mid-Office pattern.

The pattern addresses the following layers, which have been introduced in Section 4:

- Channel: multiple front-office applications can serve different channels, e.g. the website, the call centre or the front desk.
- Content: the document management system is part of the content layer.
- Business logic: the back-office applications are concerned with business logic, as is the business process engine.
- Data: The operational data store and case warehouse are part of the data layer.

Consequences: MID-OFFICE helps organizations to provide a unified, integrated face to the customer, without requiring a large-scale reorganization of back-office departments and systems. It thus resolves the need for *unified service provision* with both the need for *integrating legacy back-office applications* and the *organizational subdivision* by having a separate layer of functionality handling both data integration (the operational data store and case warehouse) and process integration (the business process engine) in a client-centric way. Furthermore, the operational data store provides the required high *availability*, *security* and *performance* by acting as a cache for back-office applications that were not designed with these requirements in mind.

The main liability of this pattern is that it could serve as an excuse to prolong the life of an outdated back-office application landscape, preventing an organization to reap the benefits of a cleaner, more manageable service architecture that removes duplications of functionality and data. Another risk is that some vendors offer integrated suites of applications as their ‘mid-office solutions’, whereas the elements of the pattern could be implemented rather independently and phased out step by step when moving to a SOA.

Known uses: This pattern and its variants are widely used by large financial institutions such as banks and insurers, who have often grown through mergers and acquisitions, resulting in a scattered and fragmented back-office application landscape. It can also be observed within Dutch municipalities [Eg06, BL08]; concrete examples include the municipalities participating in the Dimpact (www.dimpact.nl) and GovUnited (www.govunited.nl) groups.

Related patterns: MID-OFFICE contains four core elements, each of which is addressed by a pattern in its own right:

- **BUSINESS PROCESS MANAGEMENT:** to control business processes (workflows) across the borders of applications and departments;
- **CASE MANAGEMENT:** to register the progress of a client case (service usage) across applications and departments;
- **DOCUMENT MANAGEMENT:** to provide a centralized database of all in- and out-going client communication;
- **OPERATIONAL DATA STORE:** to provide a safe, reliable and 24x7 accessible data warehouse for back-office data.

Each of these patterns is worked out in more detail in our pattern catalogue.

7 Conclusions

The pattern catalogue described in this paper can provide an important asset for enterprise and business architects concerned with the design of multichannel architectures. It is not just a loosely connected set of patterns, but an integrated whole with extensive coverage of the various aspects that need to be addressed in multichannel management. Especially the comprehensive way in which the patterns were identified and mapped onto a framework of relevant design aspects may serve as an example for other pattern catalogues. Thus, the structure of the catalogue might be viewed as a pattern itself.

First reactions from practitioners have been positive. The catalogue is published online [La08] under a Creative Commons licence, to stimulate architects to reuse, improve, and extend them. We will actively monitor the development and use of the catalogue and aim to contribute its content to other existing pattern repositories.

Acknowledgments

This paper results from the ‘Kanalen in Balans’ project (‘Channels in Balance’ www.kanaleninbalans.nl) of the Telematica Instituut, a research initiative with partners from government and academia, comprising the Dutch Ministry of the Interior, the Dutch Ministry of Economic Affairs, SVB, UWV, IB-Groep, IND, VDP, the University of Twente, and Delft University of Technology. Furthermore, we want to thank the anonymous reviewers, our paper shepherd Wolfgang Keller, and all participants of the PEAM 2009 workshop for their valuable feedback on previous versions of this paper.

References

- [Al77] Alexander, C., et al., *A Pattern Language*. New York: Oxford University Press, 1977.
- [Ba07] Bayens, G.I.H.M., et al., *Nederlandse Overheid Referentie Architectuur (NORA) v. 2.0*. Stichting ICTU, The Hague, The Netherlands, 2007.
- [BL08] Bayens, G.I.H.M., Lankhorst, M.M., *De midoffice ontrafeld. Via Nova Architectura*, June 28, 2008.
- [Bo08] Booch, G., *Patterns (Handbook of Software Architecture)*, http://www.booch.com/architecture/patterns.jsp?view=kind_name%20catalog. Visited July 1, 2008.
- [Bu99] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M., *Pattern-Oriented Software Architecture: A System of Patterns*, John Wiley & Sons, 1999.
- [Eg06] EGEM werkgroep Midoffice, *Architectuur Model Gemeentelijke E-Dienstverlening: Project Referentiemodel Midoffice ten behoeve van Gemeenten*. Stichting ICTU, The Hague, The Netherlands, 2006.
- [Fi08] Fiel, E., Lankhorst, M.M., Oude Luttighuis, P.H.W.M., *De multichannel-dienstverleningscontext*. TI/RS/2008/020, Telematica Instituut, Enschede, The Netherlands, 2008.
- [Fo02] Fowler, M., *Patterns of Enterprise Application Architecture*. Addison Wesley Signature Series, 2002.
- [Ga95] Gamma, E., Helm, R., Johnson, R., Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- [HW04] Hohpe, G., Woolf, B., *Enterprise Integration Patterns*, Addison-Wesley, 2004.
- [Ko01] Koushik, S. Vasudeva, G., Galambos, G., Adams, J. *Patterns for e-business*. Mc Press, 2001.
- [La05] Lankhorst, M.M., et al., *Enterprise Architecture at Work – Modelling, Communication, and Analysis*. Springer-Verlag, 2005.
- [La08] Lankhorst, M.M., Klievink, A.J., Oude Luttighuis, P.H.W.M., Fiel, E., Heerink, L. & Van Leeuwen, D., *Kanaalpatronen – Functionele structuren voor multichannel-management*. TI/RS/2008/021, Telematica Instituut, Enschede, The Netherlands, 2008. <http://www.telin.nl/Project/Kanalen/Kanaalpatronen.htm>.
- [LB08] Lankhorst, M.M. & Bayens, G.I.H.M., *A Service-Oriented Reference Architecture for E-Government*. Chapter II in P. Saha (Ed.), *Advances in Government Enterprise Architecture*, IGI Global, 2008.
- [Mc06] McGovern, J., Sims, O., Jain, A., & Little, M., *Enterprise Service Oriented Architectures: Concepts, Challenges, Recommendations*. Springer, 2006.
- [Te07] Teerling, M., et al., *Multichannel Management – de stand van zaken*, TI/RS/2007/040, Telematica Instituut, Enschede, The Netherlands, 2007.
- [Za87] Zachman, J.A., *A Framework for Information Systems Architecture*, *IBM Systems Journal*, 26(3):276–292, 1987.