Project GRAAL: Towards Operational Architecture Alignment

Pascal van Eck *Henk Blanken* *Roel Wieringa*

Department of Computer Science
University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands
Email: {vaneck,blanken,roelw}@cs.utwente.nl

Abstract

This paper presents a framework for architecture alignment that can be positioned between approaches for software architecture, which concern software artefacts only, and strategic alignment models, which have a business focus. The framework is currently applied in case study research to find alignment patterns used in practice. First results presented in this paper indicate that the framework might yield an operationalization of strategic architecture alignment models. We also present an alignment pattern found that shows a difference between how architectures are designed at the application level and the infrastructure level. We think this difference is significant for practical alignment models.

Contents

1 Introduction

Alignment of IT architecture and business architecture is a critical success factor for modern organizations. An IT architecture that aligns with the business architecture of an organization reduces operating costs and can provide the opportunity for new product and service offerings. On

P. van Eck, H. Blanken and R. Wieringa. Project GRAAL: Towards Operational Architecture Alignment. *International Journal of Cooperative Information Systems*, 13(3):235-255, September 2004.

the other hand, a misalignment between the IT architecture and the business architecture leads to high maintenance costs and to the loss of opportunities to competitors.

Notwithstanding the importance of architecture alignment, practical guidelines for software architects to achieve alignment are still unavailable. First, current research on software architecture [2, 7, 15] focusses almost exclusively on software artefacts outside their context, that is, neglecting their role in providing the information needed by the organisations in which they are embedded. Second, research in the area of information management (e.g., [6]) focusses almost exclusively at the strategic level, without practical design guidelines for the operational level, which we take to include not only software development, but also e.g. the structure of the IT department or processes for architecture management.

In the GRAAL (Guidelines Regarding Architecture ALignment) project, the holy grail we are searching would bring together software architecture and the architecture of an organisation's information systems. Our (re)search method combines three approaches: (i) analytical research, in which we analyse frameworks and logical relationships among architecture decisions, (ii) case study research, in which we try to find out how architecture alignment is actually done in large organizations, identify best and worst practices, and formulate hypotheses, and (iii) empirical research to test these hypotheses, in which we conduct surveys of architectures and alignment relations and their evaluation by architects. GRAAL case studies focus on large-scale financial transaction processing (retail and wholesale banking and insurance) and government agencies.

The aim of this paper is twofold. First, we want to present the conceptual framework that we use to describe our case studies. In case study research, it is essential that cases are described using the same conceptual framework to enable comparing cases [24]. So far, it has been possible to describe all architectures encountered in five case studies in terms of this framework. Second, we want to present some results obtained by using the framework in two case studies. These results show how strategic alignment patterns can be observed in case studies. This will lead to an *operationalization* of strategic alignment models. The specific patterns found themselves are interesting as well. They show that in modern organizations, architecture at the application level is managed in a different way than at the infrastructure level. Consequently, a key alignment problem is the alignment of IT infrastructure services to the application needs of business processes. We explain this in detail at the end of this paper.

The structure of this paper is as follows. We start out, in Section 2, by positioning our research with respect to other approaches in architecture research. After that, Section 3 presents our framework for architecture representation, which constitutes the analytic part of our approach. In the GRAAL project, this framework is used to describe the architectures we find in a uniform way to discover alignment patterns. In Section 4, we present the first results obtained after a number of case studies have been performed. Section 5 concludes the paper.

2 Comparison with other research

We position our work between the area of software architecture (Section 2.1) as pioneered by Shaw and Garlan [15] and elaborated by the Software Engineering Institute (see e.g. [2]), and the area of IT strategy as characterised by Henderson's and Venkatraman's paper [6] (Section 2.2).

2.1 Software architecture

The difference between the area of software architecture and our approach is a difference in scope. Software architecture is concerned with the architecture of *software artefacts*, while our approach is concerned with the architecture of teh *information provisioning function* of an organization. By the information provisioning function of an organization we mean the collection of software applications, organizational unit(s), people, and procedures that are responsible for collecting, storing and manipulating all information needed by that organization. Other functions can usually be found in an organization as well, such as the research and development function, and the

	One system	System of systems
Organization	Structure of a department	Internal value chain [14]: connections between departments
Application	Application structure as aggregation of components	Application architecture: interconnected applications
Infrastructure	Infrastructure product (e.g., a DBMS)	Technical architecture: all harware, systems software and middleware and their connections

Tab. 1: Systems and systems-of-systems.

marketing/sales function. Table 1 illustrates the difference in scope between approaches in the area of software architecture and our approach. Approaches in software architecture usually focus on one system (middle column in Table 1), which is studied in detail, but in isolation. This system is a software artefact; software architecture does not study any system at the organization level in Table 1. Our approach mainly focuses on a system of systems (rightmost column in Table 1), in which the systems studied in software architecture are considered black boxes. While the aim of our approach is not to study organization theory, our approach needs to be able to describe structure at the organizational level to understand the role the application level plays in the information provisioning function.

To elaborate on this difference, consider the Architecture Tradeoff Analysis Method (ATAM, see [8]), which has the same scope as the area of software architecture in general.

- As the ATAM is limited to software artefacts, it focuses on evaluation criteria such as performance and availability. However, in practice we find other evaluation criteria as well that do not fit well in these classes. (Note however that it may be possible to decouple the ATAM from its focus on software artefacts and expand its scope towards the information provisioning function.)

- In the GRAAL approach, the information provisioning function is never treated in general, but always in relation to other, concrete business functions for which information has to be provided. In contrast, as the ATAM does consider a software artefact in isolation, it is possible to apply the ATAM when for instance developing commercially offered general purpose software of which the specific context cannot be fully known in advance.

The point above indicate the difference between the scope of ATAM and of our approach. Some overlap between both scopes can also be identified. For instance, on the one hand, the area of software architecture identifies so-called 'business drivers': the business goals that motivate the development of a software artefact. However, as we will see in the next subsection, this is only one of four ways to approach architecture alignment. On the other hand, software artefacts are components of an architecture in the scope of GRAAL, and therefore GRAAL studies the relations between these artefacts, the processes they support, and the infrastructure they are supported by.

The difference between the GRAAL project and the currently popular Model Driven Architecture (MDA, see [9]) proposed by the OMG is comparable to the difference between software architecture and our approach: the MDA is primarily concerned with software artefacts and not with the information provisioning function of an organization as a whole.

2.2 Strategic alignment

Our work can be compared to the Strategic Alignment Model (SAM) proposed by Henderson and Venkatraman [6]. The SAM is depicted in Figure 1, in which bidirectional arrows represent alignment relations. As some of our preliminary results are based on an interpretation of the SAM in the GRAAL model, we describe the SAM rather in detail before discussing the relation between our work and the SAM.

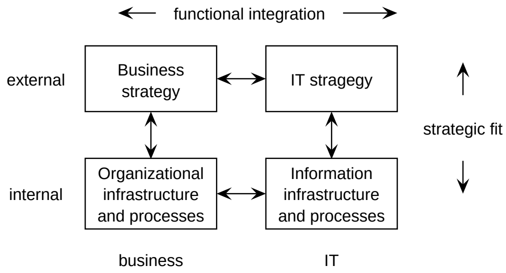
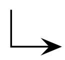
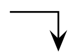
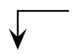
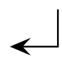


Fig. 1: Henderson's and Venkatraman's Strategic Alignment Model [6].

The SAM distinguishes four areas, along two dimensions (called 'strategic fit' and 'functional integration') that together constitute the overall strategy of an organization. The strategic fit dimension distinguishes between the external and internal domains of alignment. The external domain concerns the position of an organization relative to its competitors with respect to its product/market-combinations, partnerships, integration, etc. The internal domain concerns the structure of the organization, i.e. its departments, its processes, governance, etc. The functional integration dimension distinguishes between the information technology (IT) function and all other functions of an organization. The IT function concerns the scope, competencies, governance, architectures, processes and skills with respect to information provisioning for the business. The other side of this dimension concerns the same aspects with respect to the mission of the organization.

Perspective	Driver	Role of top management	Role of IT management	Performance criteria	Alignment approach
Strategy execution	Business strategy	Strategy formulator	Strategy implementer	Cost/service center	
Technology transformation	Business strategy	Technology visionary	Technology architect	Technology leadership	
Competitive potential	IT strategy	Business visionary	Catalyst	Business leadership	
Service level	IT strategy	Prioritizer	Executive leadership	Customer satisfaction	

Tab. 2: Alignment perspectives distinguished by Henderson and Venkatraman [6].

Henderson and Venkatraman argue that, except for very simple cases, strategic alignment always needs to involve both dimensions. In other words, except for very simple cases, strategic alignment never only concerns one (bidirectional) arrow in Figure 1, but a combination of two arrows, one horizontal and one vertical. They describe four perspectives on strategic alignment that involve both dimensions and differ by which strategy area acts as a driver, the role of top management and IT management, and performance criteria for assessing the IT function. Table 2 summarizes the four perspectives. In the arrow patterns, an arrow denotes the direction of a driving force.

The merit of the SAM is in its recognition of an external orientation of IT strategy. It is all too easy to approach internal strategy from an internal point of view only (as in the strategy execution perspective). Notwithstanding this, there are many cases in practice [1] in which the specific position of a business with respect to its IT competences enables creating a strategic advantage, which indicates IT strategy has to be present in a model of strategic alignment. However, the SAM model has several drawbacks for the practising architect. First, it is hard to apply the model in practice, as Henderson and Venkatraman do not provide an operationalisation of their model (and to the best of our knowledge, no accepted, tested and sufficiently disseminated operationalisation exists to this date). Thus, given a specific alignment case study, there are no objective, concrete criteria to determine which of the alignment perspective(s) plays a role in the case. Second, the SAM is not a constructive theory of strategic alignment: it does not provide any guidelines on how to reach specific alignment goals. This is the main difference with our approach, as the goal of the GRAAL project is to find such guidelines.

Maes et al. [12] have adapted the SAM by extending both dimensions with one extra level (Figure 2). In their framework, the internal level of the strategic fit dimension is split in two levels, called 'structure' and 'operations'. This extension is motivated by the observation that within the internal domain, operational processes first have to be designed (to determine their structure) before they can be executed. These activities have different frequencies of occurence and are managed in different ways. In the functional integration dimension, the IT level is split in 'technology' and 'information and communication' and to distinguish between pure technology (software development and maintenance, and exploitation) and information dissemination within an organization. This extension reveals a fundamental difference in motivation: while Henderson and Venkatraman emphasise that an organization's competences with respect to information *technology* is important enough to develop a model for strategic alignment that makes IT strategy explicit, Maes [11] argues that "information *itself* rather than ... technoglogy as the real carrier of value and source of competitive advantage [5]" (emphasis added), thus downplaying the importance Henderson and Venkatraman attribute to the technology itself.

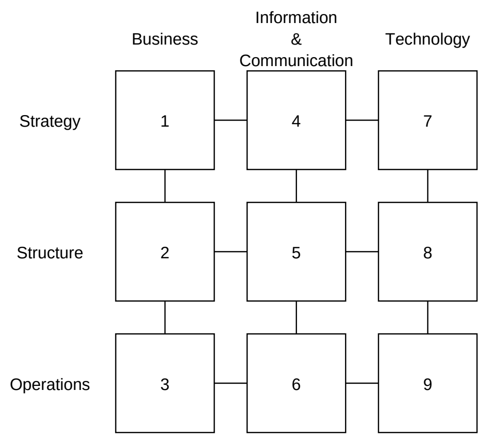


Fig. 2: The alignment model proposed by Maes et al. [12].

2.3 Software architecture in the Dutch software industry

Over the years, several practitioners in the Dutch software industry have published books about architecture alignment (see e.g. [17, 10, 20]). This work is comparable in scope with our approach and therefore the GRAAL project intends to relate our results to these publications. However, at least one important difference can be identified as well: these publications present alignment *methods* developed analytically from practical experience that describe how architecture alignment *should be* managed, while our approach aims to develop *recurring patterns* that describe how architecture alignment *is currently* managed in practice. Our result have the form "If A happens, then B is the effect". The value of this for design is that this observation can be used to make design decisions: "If you want to achieve B, you can choose to do A; and if you want to avoid B, don't do A".

3 The GRAAL framework

3.1 Service layers and refinement

In the GRAAL project, each case study is represented in a uniform way according to the GRAAL framework. In its most extended form, the GRAAL framework consists of four dimensions [18]. We give a brief overview and then discuss the elements relevant for this paper.

- Lifecycle. The process in which the architecture is planned, designed, realised and deployed.
- Aspects.
  - External: Functions, behavior, communication, dictionary.
  - Internal: Composition.
- Service layers. Components provide services to one another, and this can be strucutred into layers, as explained in more detail below.
- Refinement. The amount of detail the description of the architecture provides.

Note that unlike the other dimensions, the refinement dimension is strictly a description dimension: It concerns descriptions, not the described entities. Similar to our earlier work [23], in this paper we use a version of the framework in which only the service layer and refinement dimensions are explicit.

At the service layer dimension, we distinguish five service layers. Products provide services (have functions) to one another. We assume that service provision is structured as a directed graph that shows who delivers services to whom. This graph can itself be structured into layers, such that lower layers provide services to higher layers but not vice versa. So cycles in the service-provision graph are restricted to layers and do not cross layer boundaries. The layers and the interfaces between the layers are depicted in Figure 3. At the lowest three layers, our service layer dimension is the same as the layer pattern [3] or virtual machine pattern in software and hardware [16] architecture. The design guideline behind it is to reduce complexity by abstracting functionality in virtual machines. Software at a higher layer, e.g. a DBMS, cannot run when the virtual machine for which it was written is not available, but not the other way around. In this sense, the layers are objective: it is possible for other people than the original designer to observe them. The two highest levels are also objective: the business process layer consists of people and processes that belong to the organization that is being studied, the business environment layer consist of people and other phenomena outside the organization.

- The *business environment* consists of the value chain in which the business operates. This includes clients and client groups, suppliers, competitors, government bodies, distribution and communication channels.

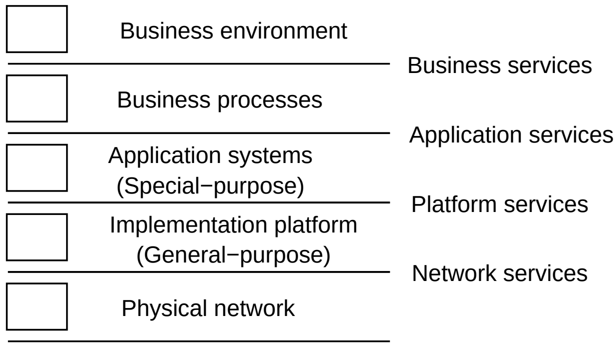


Fig. 3: Service layers in the GRAAL framework.

- The business is an organization of people and machines with a common purpose to deliver a product or service to a market. To deliver a product or service, the business has to execute *business processes*: Operational processes that respond to external and temporal events to deliver products and services, supporting processes and management processes.
- *Application systems* support or fully perform parts of the operational and other business processes.
- The *implementation platform* is the collection of standard general-purpose software needed to run the application software. It is also called "implementation platform". It ranges from operating systems, middleware, network software to database management software.
- The *physical network* consists of the physical resources that run the software platform and the application software. "Physical" means "having a size and weight". The network consists of boxes that contain metal, plastic and silicium, copper wires and other physical entities.

In Figure 3, each line between two layers generally represents a many-many mapping between entities of the two adjacent layers.

The refinement dimension (Figure 4) takes us from abstract descriptions (containing little detail) to detailed description. At the same time, it takes us from strategic decisions (taking the long term global view) to operational decisions (taking the short term detailed view).

3.2 The Strategic Alignment Model interpreted in GRAAL

The Strategic Alignment Model (SAM) can be mapped to the GRAAL framework by equating the strategic fit dimension of the SAM with our refinement dimension, and by equating the functional integration dimension with our service provisioning dimension (Figure 5). We motivate this as follows:

- The organizational infrastructure and processes are designed such that they serve the business strategy, and their description should *elaborate* and *explain* their relationship with strategy. In this sense, (the description of) strategic fit equals refinement: strategy is elaborated and explained in terms of the tactical and operational choices made to execute the strategy. Note that in all four perspectives described by Henderson and Venkatraman (Table 2), the external domain is always the driver of the internal domain and not the other way around.
- As the IT function of an organization is a supporting function for the business functions, functional integration can be equated with service provisioning in the context of this paper. Note that although our stack of service layers is directed (a lower layer provides services to a higher layer and not the other way around), it is possible that a lower layer drives decisions

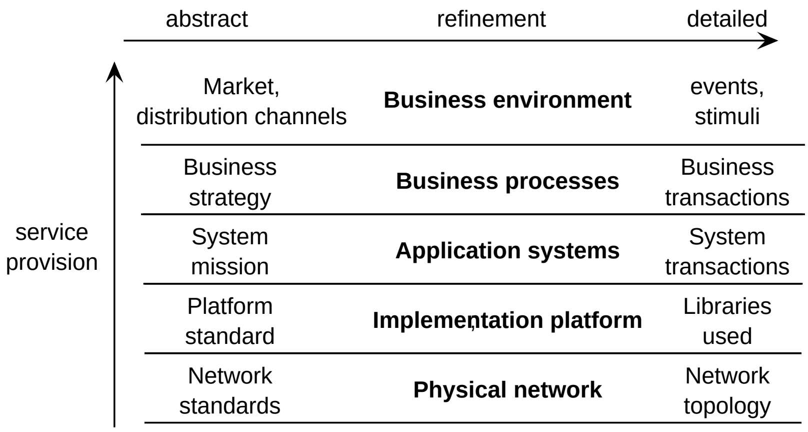


Fig. 4: Refinement.

with respect to the design of a higher layer. This is similar to the four perspectives depicted in Figure 1, in which horizontal arrows go either way.

The four areas of the SAM (Figure 1) can be placed in our dimensions as follows (see Figure 5):

- The *Business strategy* area describes the position of a business relative to its competitors with respect to its product/market-combinations and its place in the value chain. This is precisely the contents of the environment layer in the GRAAL framework.
- The *IT strategy* area describes the position of a business relative to its competitors with respect to its IT competences, IT outsourcing, etc. Notwithstanding the external orientation of this area, it is important to note that *information technology itself is internal in that it serves the primary business of an organization*[1]. Therefore, this area is in the lowest three layers of the GRAAL framework, and not in the upper two. Due to the problems of the SAM with respect to operationalization mentioned in Section 2.2, it is difficult to determine exactly whether IT in the SAM includes aspects of software applications that are not of a technological nature, such as for instance requirements engineering and management.
- The *Organizational infrastructure and processes* area describes the processes and organization of a business. In the GRAAL framework, the business processes layer, which may include descriptions of the business structure, serves the same purpose. Again, as the border between information technology and its applications in the SAM is not exactly clear, it may be the case that this area extends for some part in the applications layer.
- The *Information infrastructure and processes* area is placed at the same layers as the IT strategy area, and the same remark with respect to the exact borders of the area applies. This area is a refinement of (the description of) the IT strategy in the sense that it elaborates on the IT strategy.

The reinterpretation of the SAM in terms of the GRAAL framework is depicted in Figure 5. In this figure, bidirectional arrows between strategy areas (the four boxes) denote alignment relations.

Note that the original SAM (Figure 1) appears in the GRAAL framework (Figure 5) mirrored with respect to the diagonal axis from upper left to lower right. This means that the arrows that

[1] In the case of a business in the software/hardware or content industry, there are in fact two kinds of information and IT: the information and IT that is on sale, and the information and IT that supports selling the first kind. In this paper, information and IT always refer to the second kind.

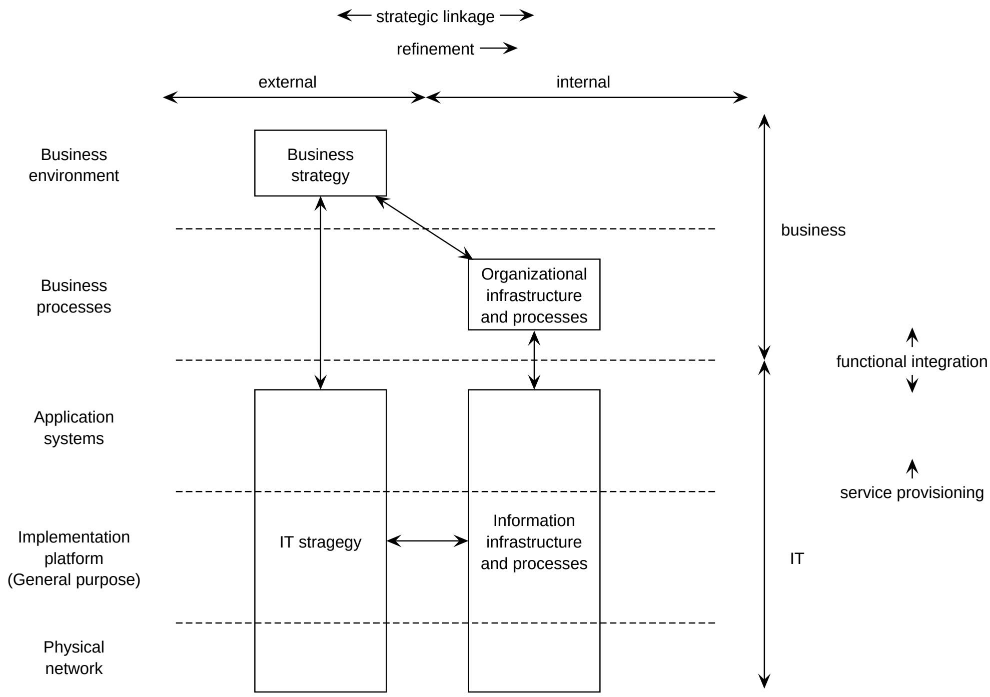


Fig. 5: The Strategic Alignment Model (SAM) in the GRAAL framework.

appear in Table 2 have to be mirrored as well to recognize the four alignment perspectives. Figure 6 depicts these alignment perspectives as the appear in the GRAAL interpretation of the SAM.

It is possible to reinterpret the model by Maes et al. [12] in terms of the GRAAL model as well. This reinterpretation is depicted in Figure 7. The numbers refer to the numbers in Figure 2.

4 Results

4.1 Research method

Since the start of the GRAAL project, we have completed three cases: one in a small Internet startup [23] and two in government organizations. (A fourth and fifth case study are currently being carried out). In this section, results are presented based mainly on the two governmental case studies. The two government organizations are a large one at the national level (which we will call BIG) and a medium-size local government office (which we will call LGO). Our results lead to hypotheses for further study, which are presented in Section 4.4.

The case studies have been carried out primarily on the basis of project documentation. For each case study, project documentation was obtained which was studied off-site. After that, for each case study, a document was prepared describing observations regarding organizational structure, the architecture development process and architecture documentation, an analysis of the alignment relations found, hypotheses and further research questions, and recommendations. If needed, extra documentation was obtained. This report was discussed with architects from the respective organizations to identify factual errors and present recommendations. In project documentation, alignment relations appear in the form of motivations of design decisions: the architecture of one layer is motivated by goals at a higher layer that it helps to bring about. Therefore, to find guidelines for architecture design decisions, we have traced back all design decisions.

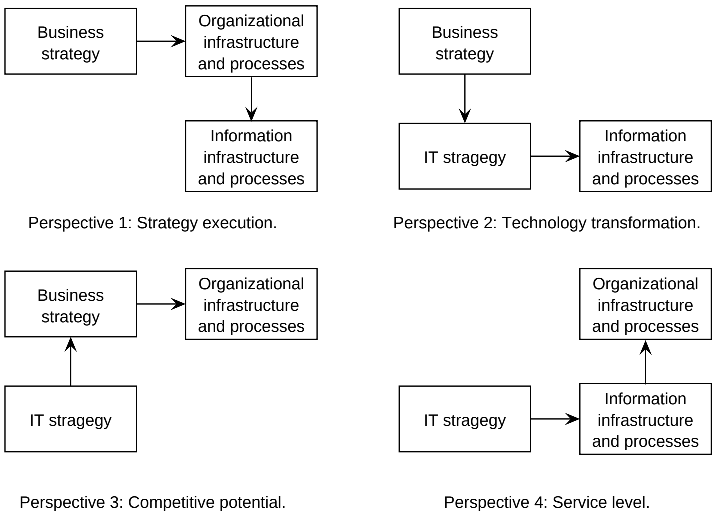


Fig. 6: Alignment perspectives distinguished by Henderson and Venkatraman [6]. Horizontal and vertical axes interchanged compared to Figure 1 to fit in the GRAAL framework.

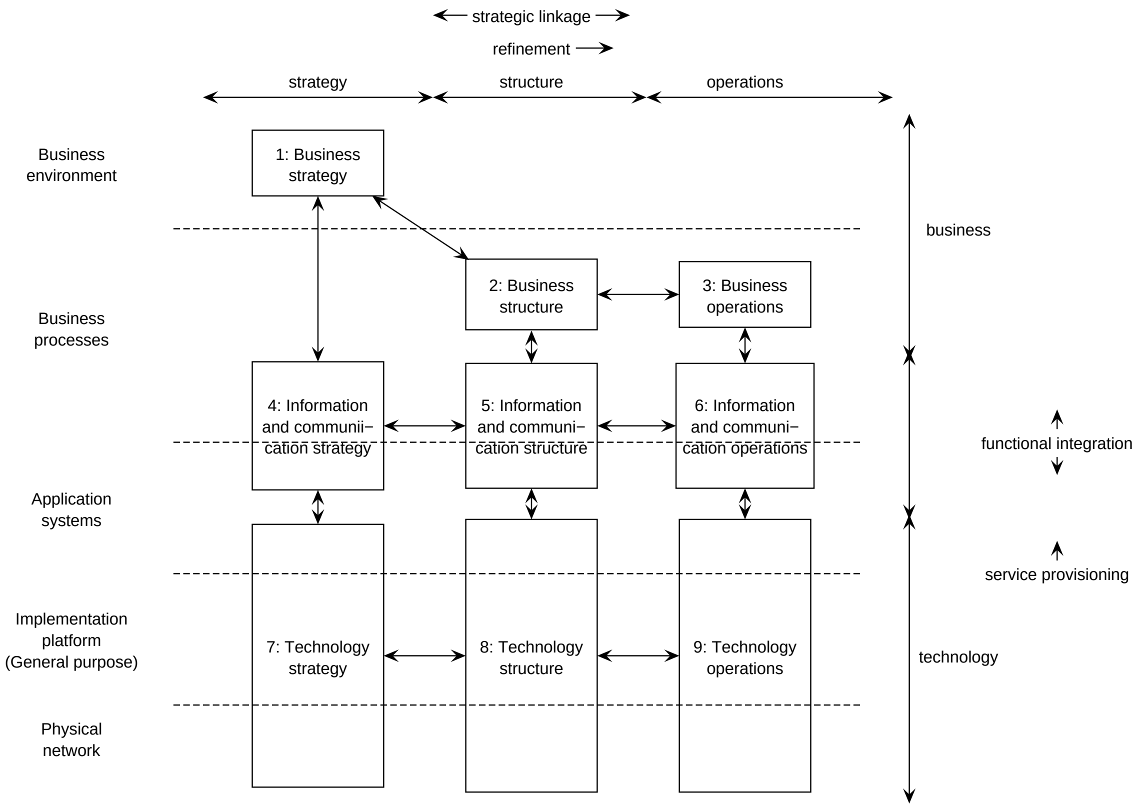


Fig. 7: The alignment model proposed by Maes et al. [12] in the GRAAL framework.

In the rest of this section, we first report some observations that can be explained by Conway's Law, which will be introduced below. After that, we present our main result: a difference we have observed between alignment at the application layer and at the infrastructure layer. We describe our evidence for this difference and explain it in terms of the Strategic Alignment Model of Henderson and Venkatraman. This observation leads to two hypotheses for further research, which are presented at the end of this section.

4.2 Conway's Law

Conway's Law [4] states that the architecture of a system is isomorphic to the communications structure of the organization that has designed the architecture. For the LGO case, no information on the structure of the organization (a small project team from an outside contractor) was available. For the BIG case, the situation was different. During the time this case was carried out, the IT department of BIG (which we call BIGIT) was reorganized, which enabled us to observe the effect of organization structure on the architectures that it produces. The old structure of BIGIT was as follows. At the top level, three sectors were distinguished, one for each software lifecycle phase:

- A systems development sector. This sector was subdivided in five departments, three four three different client groups, one general department and a department responsible for the architecture of the systems development sector itself (which can be viewed as a fourth client group). *Each department had its own architects*, as well as its own account managers, developers, testers, etc.
- A systems integration sector. This sector was subdivided in five clusers, each specializing in a particular technology: middleware, platforms, networks, end user environment, and telematics. Again, each department had its own architects.
- A systems management and exploitation sector, subdivided in three groups: management, exploitation and service delivery.

The new structure consists at the top level of ten sectors distinguished by their function, e.g. account management, procurement, exploitation, service delivery, etc. One of these new sectors is the sector 'architecture', in which all architects can be found (business architects, application architects and infrastructure architects). In the new structure, *architects are closer together, but no longer organized by client group*

We observed that in this case, the structure of the architecture documentation was isomorphic to the structure of the organization that designed this documentation. In the old structure, the application-layer architecture (in the sense of the GRAAL framework) was designed by the systems development sector, while the architecture at the level of the software platform and physical network was designed by the systems integration sector. This complies with Conway's law, as the documentation of a system is itself a system, and it is structured similar to the structure of the designing organisation (both are divided in separate parts along the same dividing line and are separate as our study found out).

In the new structure, all architects can be found in one department, but this department itself is divided in three groups: business-level architects, application-level architects and software-platform/physical network architects (in GRAAL terminology). By Conway's law, we can predict that also in the new structure, the application-layer architecture and the software-platform/physical network architecture will be developed separately and independent from one another.

As a consequence of both the old and new structure and Conway's law, systems that end users see can be divided in two groups: systems developed in-house at the application layer (applications and data management systems) and standard commercial systems (decision sypport systems, ERP, office automation, document management) at the software platform layer. End users, however, perform tasks for which this difference is not relevant. Thus, the IT organization as a whole delivers a set of services that contains a non division which is not functional for end users. We

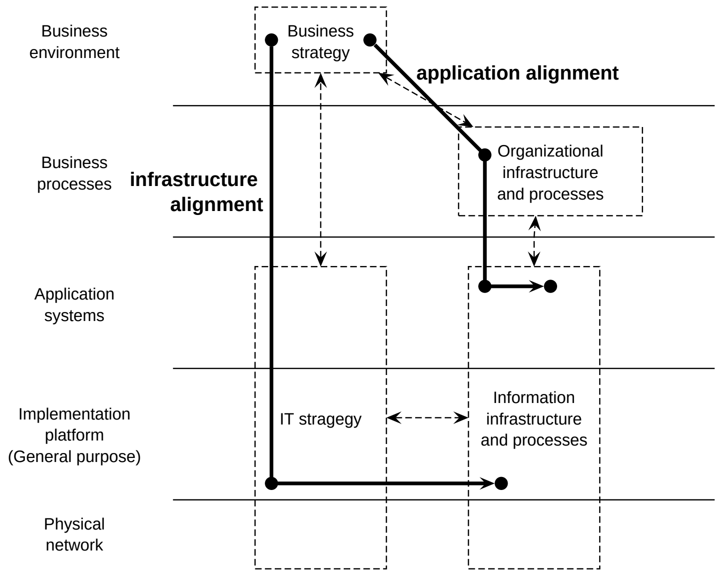


Fig. 8: The alignment framework. The bullets represent documents produced in the alignment process, discussed later.

therefore recommend to carry those parts of the software platform that are directly used by end users over to the application layer.

4.3 Strategic Alignment: Application Level and Infrastructure Level

In both government cases, we found that design decisions at the application layer are motivated differently compared to design decisions at the software platform / physical network layers. (In the third case, no such difference has been found. In this case, the same small team designed the architectures at all layers and there was no legacy to take into account.)

At BIGIT, alignment between applications and business was done by first designing the business processes in accordance with the business strategy, and then developing application support to fit these processes. Alignment between infrastructure and business was done by first formulating an IT strategy in accordance with the business strategy, and then implementing the IT strategy in the infrastructure domains. These two paths do not necessarily lead to points that are in harmony with each other. At BIGIT, there was considerable tension between the two, in terms of mismatch between application requirements on infrastructure, and services offered by the infrastructure.

In the next two sections we discuss the two alignment paths at BIGIT in more detail. After that, we briefly discuss strategic alignment in our other case study.

4.3.1 Application alignment relations at BIGIT

Application alignment at BIGIT takes place in three steps, that produce three documents (Figure 8). Starting from the business strategy (the bullet at the tail of the application alignment arrow), a design of the business processes is produced (the second bullet). This is the basis for a document describing the architecture of the entire application layer (third bullet), which in turn is the basis for the architecture of each individual application (fourth bullet).

At BIGIT, the application layer is split in two, an information system layer and what we will

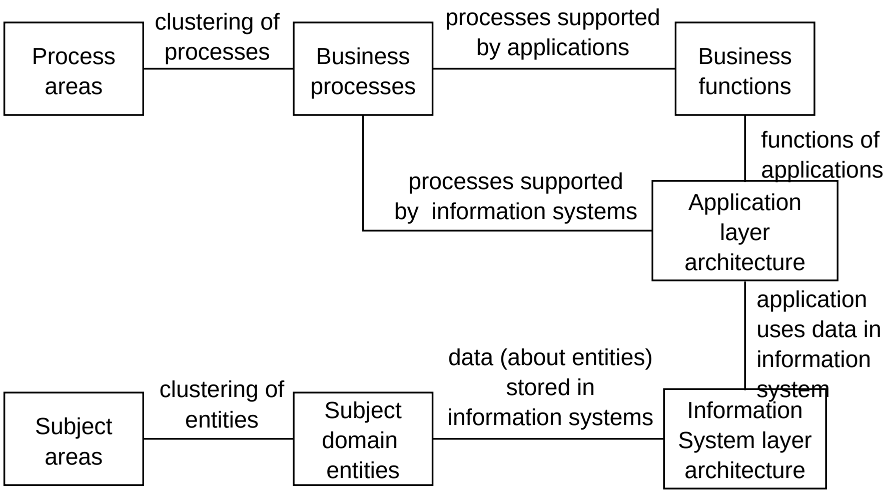


Fig. 9: Application alignment relations.

call a "pure" application layer. The information system layer contains the data storage systems and the pure application layer contains the data manipulation systems. Pure applications are stateless; all state is maintained by the information systems. The term "application" is thus ambiguous because it can refer to pure, stateless applications only, and to information systems plus stateless applications jointly. This ambiguity turns out not to be harmful and we will provide disambiguation where needed.

Figure 9 shows the application layer architecture descriptions produced at BIGIT for one of their customers, which is another department of BIG that we call D. We only explain the major descriptions.

- A list of *business functions* of D is made. For each function, its contribution to the business goal of D is described. Functions are organized into a function refinement tree with the business mission of D at its root and elementary functions at its leaves.
- A description of *subject domain entities* of D is made. The subject domain of a business is the part of the world about which the business contains data. (Note that D is an information-processing organization.) The subject domain description is cross-checked with the business function descriptions by checking whether the entities about which a function needs data, are described in the subject domain description. (This cross-check is not represented in Figure 9.)
- A description of end-to-end *business processes* of D is made, leading from a customer of D to a result for that customer of D. This is cross-checked with the function refinement tree (does every process support a function and vice versa?).
- The descriptions of functions and processes are used to identify an *application layer architecture.* This is a list of applications and for each application, its role in one or more business processes, and its contribution to business functions.
- The application layer architecture and subject domain description are used to design an *information system architecture.* This is a list of information systems and for each information system, the data it stores and the applications it supports.

These descriptions have been produced using a method called Panfox [17], which is itself based on Information Engineering [13]. Observe that application alignment is defined at BIGIT for all of the external aspects of the aspect dimension of our GRAAL framework (see Section 3.1):

- Functions (business functions in Figure 9).
- Behavior (business processes and process areas in Figure 9).
- Communication (the application layer and information system layer architectures, and process areas in Figure 9).
- Dictionary (subject domain entities and subject areas in Figure 9).

For each individual application of D, BIGIT executes a systematic process to define its internal architecture based on the application layer architecture. Basically, the application layer architecture provides the context diagram for each individual application and information system. Application architecture design is software architecture design and due to space restrictions we will not discuss this here.

To summarize, at BIGIT, the application architecture is derived via the organizational infrastructure and processes from the business strategy of BIG. In terms of the Strategic Alignment Model (SAM) of Henderson and Venkatraman, this is the *strategy execution* perspective.

4.3.2 Infrastructure design arguments at BIGIT

The customers of BIGIT are all other departments of BIG, of which D is only one. The infrastructure architecture is motivated in terms of four kinds of phenomena that do not refer to the needs of D but of BIG as a whole.

- *Business goals* of BIG. Examples goals are the improvement of accessibility of BIG's services, the improvement of customer-friendliness and the facilitation of mobility of BIG's employees.
- *Business problems* experienced at BIG. Example problems are the high cost of maintenance, the large number of systems for which users must remember passwords, and the lack of OS/390 experts.
- *Current systems* at BIG. These range from ERP systems used to the range of operating systems in use (and there various versions), all classified according to the infrastructure domains that BIGIT distinguishes (e.g., ERP, operating systems, ...).
- *The current technology market trends* for off-the-shelf applications, office software, document management software, middleware, telematics software, etc., again organized according to technology domain.

These considerations transcend not only the individual business processes at D. They also transcend D. In fact, D must define a business strategy that agrees with BIG's strategy. The infrastructure alignment arrow in Figure 8 starts from a BIG business strategy, which covers all aspects of the business, including some very general IT aspects. This is then specialized into a strategy for the IT infrastructure only. This is in turn refined into a description of the next version of the infrastructure architecture for BIG, and the acquisitions of IT that follow from this. We observed a number of related alignment failures at BIGIT, that can all be traced to the following phenomenon: Each infrastructure domain is a technical knowledge area that requires several years of study followed by constant attention to the trade press to understand. So for each domain, there is a technical specialist gathering domain knowledge. It proves to be very difficult to relate the insights of all technical specialists to one another, let alone to managers who do not have a technical background.

To illustrate infrastructure design at BIGIT, we show in Figure 10 a fragment of a goal-directed argument that we encountered at BIGIT. This is a rational reconsruction of part of the reasoning in the infrastructure documentation. Some of the goals are ill-defined, and others are redundant. In Figure 10, the goals in bold italics type are goals at the level of the business strategy of BIG. The goals in normal type are IT strategy goals of BIGIT that contribute to the goals at the business level.

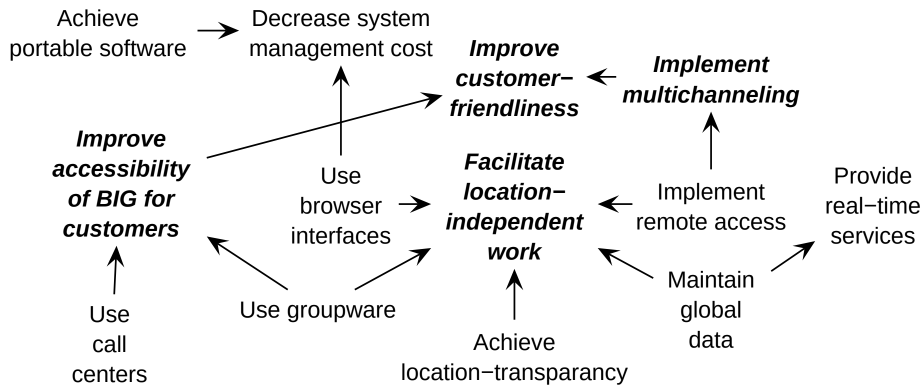


Fig. 10: Goal graph. Each node represents a goal chosen by BIG. $A \rightarrow B$ means that A contributes to B, in other words that B is a reason to choose A.

To summarize, at BIGIT, the infrastructure architecture is derived via the IT strategy from the business strategy. In terms of the SAM, this is the *technology transformation* perspective.

4.3.3 Strategic alignment at LGO

At the local government organization (LGO, our second case study), we identified a similar difference between design decisions at the application level and the infrastructure level:

- The architecture at the business processes layer and application layer is motivated extensively and explicitly in terms of how these architectures contribute to the business mission of the organization studied. The line of reasoning in the project documentation started with the mission of the organization studied. This mission was refined into a set of "service delivery principles", which do not contain any IT-related terms. The service delivery principles were refined themselves in numerous "consequences", most of them referring to processes and management information, and a few of them referring to properties of the IT needed. In terms of the SAM, this is the *strategy execution* perspective.

- The architecture at the software platform and physical network infrastructure is for a large part motivated by reference to other projects that were carried out concurrently (but by other teams). The design choices and their motivations from these projects were not available to us. However, the documentation of the project studied describes a number of advantages of the architectures designed by the other projects. These advantages do not refer to the business mission of the organization studied, but to its relation with software vendors (the architecture is as open and modular as possible to avoid vendor lock-in). Thus, the architecture at this level seems to be motivated by the relative position of the IT department in the value chain. In other words, it is motivated by the IT strategy of LGO. In terms of the SAM, this is the *service level* perspective.

4.4 Discussion

In both cases, we have observed that the design of the architecture of the infrastructure level is motivated in a different way than the architecture at the application level, and that the infrastructure level architecture is much less motivated by the business processes it supports (indirectly via the application level). This may be due to the fact that an infrastructure is never designed for one specific group of users, of which the requirements can be fully determined. On the contrary, we argue that an IT infrastructure is designed for a large part without knowing the requirements of its users. This has important consequences, e.g. application designers need to be aware of the fact that the infrastructure is largely without their design charter.

Given these findings, we state the following hypotheses:

H1 The IT infrastructure of a large organization is designed at a time when most of its users are not known. The design of the infrastructure is therefore not motivated by user needs, but by the IT strategy of the organization.

H2 Alignment at the application level is motivated by *both* end user needs *and* by the features of the currently available infrastructure.

The consequence of the first hypothesis is that factors such as technological developments in the IT market and availability of personnel for a specific technology in the labour market have a much greater influence on the design of the infrastructure than user needs. These factors are beyond management control. Given this consequence, we argue that application designers should view their task as aligning the needs of end users with the infrastructure services provided [19], and not as deriving infrastructure needs from end user needs. How this is to be done is a topic for further research.

The second hypotheses expresses the importance of taking both end user needs and features of the currently available infrastructure into account in management of the application level infrastructure. Only taking end user needs into account leads to an application architecture that most probably cannot be realized using only the available infrastructure. Investments in new infrastructure can be made, but new infrastructure tends to extend (not replace) the available infrastructure, leading to unnecessary maintenance costs.

Designing an application architecture purely from end user needs is a relatively well-understood area. Design guidelines for this task have been identified and published elsewhere ([22], Chapter 18). However, designing an application architecture by taking into account the features of the infrastructure is much less understood. The guidelines identified by the Software Engineering Institute [21] for building systems from commercial components may form a good starting point.

5 Conclusion

We have presented a framework for architecture alignment that can be positioned between approaches for software architecture (that concern software artefacts only) and strategic alignment. This framework is used as a foundation for case study research to find alignment patterns used in practice. Our first results indicate that our approach might yield an operationalization of a strategic architecture alignment model. Moreover, we have found a difference between how architectures are designed at the application level and the infrastructure level. Generalizations such as the two hypothesis listed above can be used by designers to guide their architecture decisions.

In future research, we will first apply our framework to more case studies. This should result in more and more precise hypotheses. After that, we will validate our hypotheses by structured surveys. The ultimate goal of this research is to find guidelines that will assist the practicing architect in aligning architectures at all levels.

Acknowledgements

The authors wish to thank the organizations that have provided access to their architecture project documentation. The GRAAL project is jointly funded by Telematica Instituut and CTIT.

References

[1] L. M. Applegate, R. D. Austin, and F. W. McFarlan. *Corporate Information Strategy and Management: The Challenges of Managing in a Network Economy.* McGraw-Hill/Irwin, 6th edition, 2002.

[2] L. Bass, P. Clements, R. Kazman, and K. Bass. *Software Architecture in Practice.* SEI Series in Software Engineering. Addison-Wesley, 1998.

[3] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. *Pattern-Oriented Software Architecture: A System of Patterns.* John Wiley & Sons, 1996.

[4] M. E. Conway. How do committees invent? *Datamation*, 14(4):28–31, 1968.

[5] R. Glazer. Measuring the value of information: the information–intensive organization. *IBM Systems Journal*, 32(1):99–110, 1993.

[6] J. C. Henderson and N. Venkatraman. Strategic alignment: Leveraging information technology for transforming organisations. *IBM Systems Journal*, 32(1):4–16, 1993.

[7] C. Hofmeister, R. Nord, and D. Soni. *Applied Software Architecture.* Addison-Wesley, 2000.

[8] R. Kazman, M. Klein, and P. Clements. ATAM: Method for architecture evaluation. Technical Report CMU/SEI-2000-TR-004, Carnegie Mellon Software Engineering Institute, 2000.

[9] A. Kleppe, J. Warmer, and W. Bast. *MDA Explained. The Model Driven Architecture: Practice and Promise.* Object Technology Series. Addison-Wesley, 2003.

[10] A. Ligthart and J. Vis, editors. *Applicatieontwikkeling onder architectuur.* Ten Hagen en Stam, 2002. In Dutch.

[11] R. Maes. Reconsidering information management through a generic framework. PrimaVera Working Paper 99-15, University of Amsterdam, Department of Accountancy and Information Management, 1999.

[12] R. Maes, D. Rijsenbrij, O. Truijens, and H. Goedvolk. Redefining business – IT alignment through a unified framework. PrimaVera Working Paper 2000-19, University of Amsterdam, Department of Accountancy and Information Management, 2000.

[13] J. Martin. *Information Engineering.* Prentice-Hall, 1989. Three volumes.

[14] M. E. Porter. *Competitive Advantage.* The Free Press, 1985.

[15] M. Shaw and D. Garlan. *Software Architecture: Perspective on an Emerging Discipline.* Prentice-Hall, 1996.

[16] A. S. Tanenbaum. *Structured Computer Organization.* Prentice-Hall, third edition, 1990.

[17] W. van der Sanden and B. Sturm. *Informatie-architectuur: de infrastructurele benadering.* Panfox Holding, 1997. In Dutch.

[18] P. van Eck, H. Blanken, M. Fokkinga, P. Grefen, and R. Wieringa. A conceptual framework for architecture alignment guidelines. project GRAAL WP1 whitepaper. Technical report, Centre for Telematics and Information Technology, 2002.

[19] P. van Eck and R. Wieringa. Requirements engineering for service-oriented computing: A position paper. In J. Gordijn and M. Janssen, editors, *Proceedings of First International Workshop on e-Services, ICEC'03, Pittsburg, PA, USA*, pages 23–28, 2003.

[20] R. Wagter, M. van den Berg, J. Luijpers, and M. van Steenbergen. *DYA: snelheid en samenhang in business- en ICT-architectuur*. Uitgeverij Tutein Nolthenius, 2001. In Dutch.

[21] K. Wallnau, S. Hissam, and R. Seacord. *Building Systems from Commercial Components*. SEI Series in Software Engineering. Addison-Wesley, 2001.

[22] R. J. Wieringa. *Design Methods for Reactive Systems: Yourdon, Statemate, and the UML*. Morgan Kaufman, 2003.

[23] R. J. Wieringa, H. M. Blanken, M. M. Fokkinga, and P. W. P. J. Grefen. Aligning application architecture to the business context. In J. Eder and M. Missikoff, editors, *Advanced Information Systems Engineering. 15th International Conference, CAiSE 2003, Proceedings*, volume 2681 of *Lecture Notes in Computer Science*, pages 209–225. Springer-Verlag, 2003.

[24] R. K. Yin. *Case Study Research: Design and Methods*. Sage Publications, third edition, 2003.