

## 9.3 Application Services Library, adapted to the IT-services world of the future

*For the last five years, the Application Services Library (ASL) has served as a framework for application management. But is it still “future proof”? Remko van der Pols analyzes ASL in the light of current IT developments and examines the possible consequences for the next version of ASL.*

### INTRODUCTION

In 2002, the Application Services Library was launched into the public domain as a framework for application management. The framework is promoted and supported by the ASL Foundation (now the ASL BiSL Foundation<sup>1</sup>), and sponsored by both IT service providers and user organizations, who benefit from sharing their best practises and using a knowledge platform for application management. The adoption of ASL in the market was quite fast, and it was implemented in many organizations, primarily in the Netherlands.

After five years, the question arose as to when a new version of ASL would be launched. One of the reasons for this question was the publication of ITIL V3. One of the ASL strategies was to create a stable standard in order to allow organizations to benefit from their investments in their processes and best practices. But it was now necessary to check whether the ASL framework still provided the answers to both present and future questions.

### Goal

In this article we describe the main features of ASL version 1.1. This next release of ASL will be published as a new book. For the present, we use the name ASL 1.1, to demonstrate that the new version contains several large changes, and yet the framework does not radically change. This fits into the philosophy of taking an evolutionary approach to improve in small steps, whilst also protecting current investments in good practices.

The biggest change in ASL is the way in which it positions itself and other frameworks in the current dynamic world of ICT services. This change in point of view has a great impact, not so much in the structure of ASL version 1.1, but in how application management and processes will fit and should be implemented in the future world of demand and supply (delivery).

### Structure article

After the introduction, the article starts with an analysis, addressing the current strengths and weaknesses of ASL. This analysis leads in to a discussion of several changes for the new version of ASL, and ends with conclusions in the last section.

<sup>1</sup> The Dutch organization that manages and develops the Applications Services Library (ASL), a framework for application management and the Business Information Services Library (BiSL), a framework for information systems management.

1

2

3

4

5

6

7

8

9

10

11

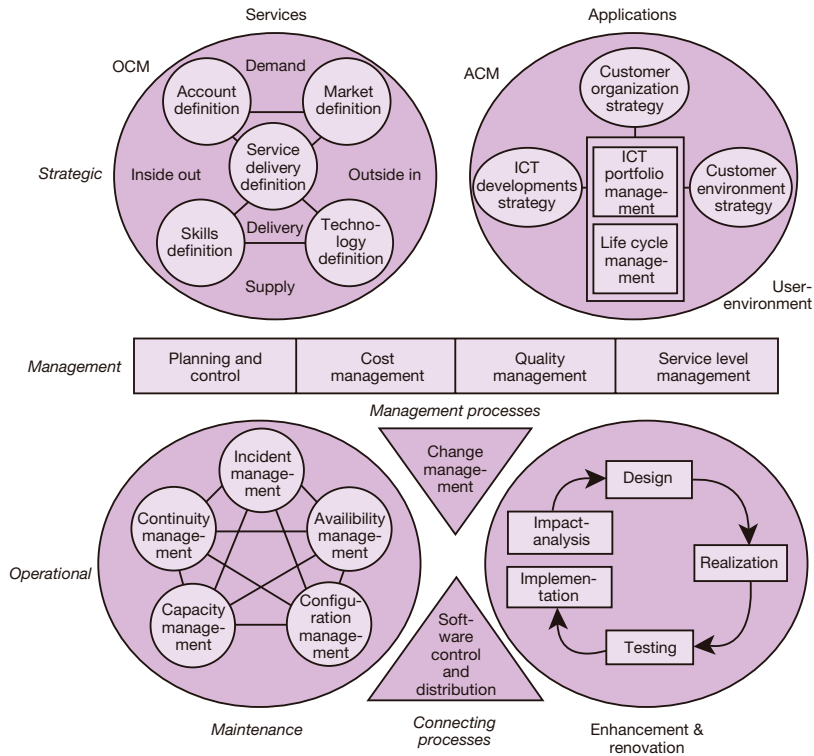


Figure 1 ASL model

### Reservations

At the time of writing this article (December 2007) the development of the next version of ASL was yet to be finalized. Most of the remaining elements deal with minor issues, such as the names of processes and clusters.

Not all remarks and issues will be discussed here, but they will be included in the final book. In this article, the main lines and major issues and changes will be described, and in some places we will also address possible solutions.

### STRENGTH AND WEAKNESS ANALYSIS

Two sources of information have been used for the development of ASL version 1.1:

- The ASL-BiSL Foundation has collected various remarks and questions on version 1.0, and there has also been a call for remarks. This issue list was one of the sources.
- A strength and weakness analysis has also been executed, and this fundamental analysis was a source for the larger changes.

The issue list and the analysis were well aligned.

#### Remarks from the issue list

The issue list was quite extensive and addressed many design decisions that were not completely clear. Many issues addressed the implementation of ASL in organizations.

The most frequent issues were:

- The alignment of ASL, ITIL and BiSL. The need for an 'integral' model or a description of how the frameworks interface.

- Questions about security management (this is one of the topics covered in the continuity management process, but it needs to be better addressed).
- Remarks about the names of processes, such as implementation, software control and distribution, and cost management.

#### Analysis:

- Most of the remarks were detailed, and were deemed to be valid and useful. The issues showed a large degree of consistency.
- Some remarks were possibly due to a lack of understanding, leading to a review of the clarity of the relevant material.
- Some remarks were related to the trend that is described in the next section.

#### Evaluation of the key concepts and structure of ASL

The structure of ASL version 1.0 contained six clusters of processes (Figure 1), with 25 processes. ASL version 1.0 also addressed several issues and had several key concepts:

- the importance of having effective service levels and an orientation on responsibility for results
- a pro-active approach towards innovation: not only from the service provider's point of view (Organisation Cycle Management), but also from a user perspective, taking a long-term view of the applications (Application Cycle Management, life cycle management)
- the service team concept as a means to integrate services, for instance, application management and infrastructure management
- differentiating between external quality (as described in service levels) and internal quality (a technical or internal view on quality)

Several conclusions were made in the strength and weakness analysis, and were confirmed by the remarks on the issue list. The most important conclusions are:

- The concept of six process clusters with underlying processes is considered, by most people, to be both useful and helpful. In the past, some questions were raised about the number of processes (26), but this question no longer arises. ITIL V3 has a similar number of processes. The concept of process clusters also created the opportunity to implement at a cluster level, rather than at a process level.
- ASL had anticipated the growing trend of information chains (organizations connecting their information processes and information flows). Managing these information chains is more complex than regular information processes, because more independent organizations are required for creating and maintaining such a flow. This causes extra complexity with regard to manageability.
- ASL did not have any technology-dependent processes (e.g. workflow management) and avoided addressing 'hypes' (like components) which are now considered outdated. The approach of creating a theoretical underlying object model, independent of technology and solution, made the model of ASL relatively sustainable. In hindsight, we considered this to be a good decision.
- We learned that the common practice is to start implementing the processes on the bottom level and to move upwards. The 'strategic' processes (the processes on the top of the model) were initially considered to be innovative, but are now regularly implemented.
- ASL made a distinction between external quality (customer perspective) and internal quality (application management perspective). SLAs had to be considered as a specification of external quality, which should be primarily filled in by the customers.

1

2

3

4

5

6

7

8

9

10

11

These messages have been important and were new when they were first implemented. The relevance of these messages is still high, but they are now generally accepted as being common sense.

### Sustainability of ASL version 1.0

The final question discussed was whether the ASL model is sustainable in the light of developments in the market. Many of the actual trends have been anticipated in the first version of ASL, but there is one very important trend remaining that necessitates a considerable change to the model. This trend is the segmentation and componentizing of ICT-services.

## COMPONENTIZATION OF SERVICES

### Reasons for componentization of services

There are many reasons for this segmentation, including:

- outsourcing
- growing importance to the business
- standardization and reuse
- variety

#### 1. Outsourcing

Outsourcing and “professionalization” has led to a gap between suppliers (the ICT-organizations) and a demand-organization. No longer is the (internal) ICT-department the place where both delivery and ICT-strategy are positioned.

A further “componentisation” of ICT service can be found on the supply side. The distinction between application management and infrastructure management has also become very common.

This can be easily explained by the fact that the business driver of infrastructure management is considered to be operational excellence, while application management increasingly depends on customer intimacy (or at least customer process intimacy). These three domains were originally introduced by Maarten van Looijen (1998), in his management model.

#### 2. Growing importance

Over the last few decades, the importance of ICT (for the business) has been growing. In many organizations, ICT can be considered as a business process, rather than something that is supporting the business (e.g. a bank).

The importance of the financial information systems is so great that, in many situations, the CFO will insist on managing it (directly) and will avoid sharing power with the CIO. The same can be said for the business information systems within the primary processes of the organization. The result is that many organizations will have several demand organizations (customers). This means that several ‘buyers’ within the business will deal with ICT-services. This also leads to componentization of ICT-services.

#### 3. Standardization and reuse

The huge growth of ICT within organizations leads to more standardization, reuse and shared solutions, and the use of many specialized solutions. In the past, a single supplier could deliver everything, now this is almost impossible.

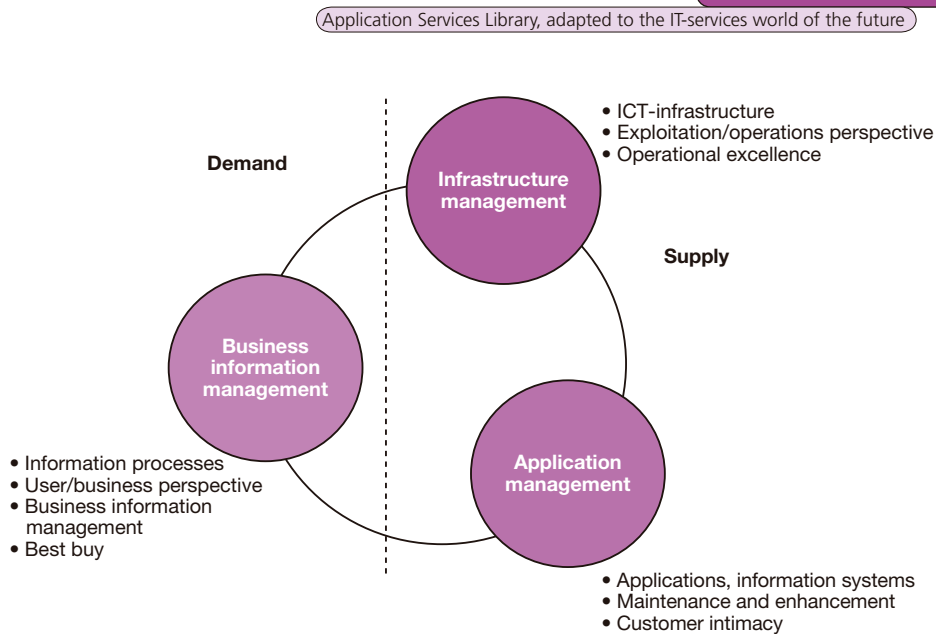


Figure 2 Management model

The consequence for application management organizations is that they have to define their core competences. They have to choose:

- which technology to use and what kind of technology can be delivered, e.g. workflow management, content management, SOA, web-based technology, mainframe development, SAP
- the segment of the market, e.g. middle and small companies, finance, government or local government, industry
- the form of delivery, e.g. projects, services, delivering man-craft

The infrastructure domain also deals with similar choices.

#### 4. Variety

There are an increasing number of ways in which services are managed, and in which services are delivered. In the past, solutions were either custom-made or standard ('off the shelf'). Now there are many situations which fall in between these two options, and also many more extreme solutions. Examples of this variety in application management include:

- building (or maintaining) separated small and completely standard solutions/components (building blocks)
- creating large customizable application platforms
- application integration
- building or maintaining applications containing standard solutions and lots of bespoke code, maintaining solutions (packages) built by others
- building complete bespoke applications

The business model or financial model may also be completely different (for instance, based on time and materials, fixed price per service, users or usage).

#### Consequences for the management of ICT-services

A logical result is that, in order to deliver ICT services normally, more than one ICT organization will be required. In normal situations, the landscape of ICT suppliers and demand organizations within a business organization will appear as it does in figure 2.

1

2

3

4

5

6

7

8

9

10

11

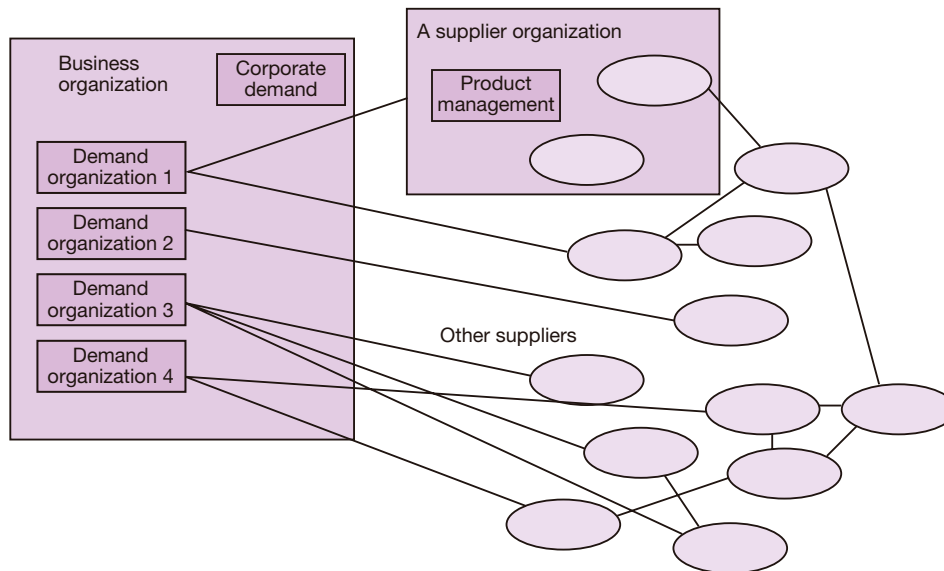


Figure 3 Modern demand and supply constellations

This ongoing development has a big effect on application management:

- Application management processes get more distinct implementations.
- Managing the ICT services will be a very complex process if some basic assumptions are not changed.

#### Application management gets many different implementations

Many different types of application management are possible:

- building and maintaining highly standardized components
- building and maintaining application platforms, such as large ASP-systems (with underlying infrastructure) or packages
- integrating and maintaining applications, built of standardized components with bespoke additions
- customization of application platforms and packages, e.g. SAP
- building and maintaining complete bespoke applications
- systems integration

*Sometimes delivery of a working solution will require eight different ICT organizations, sometimes just one. It may be difficult, even impossible, to design integral processes, because, for example, four of the eight ICT organizations might have (internal) processes, serving many other customers. Some ICT organizations might act as a systems integrator, others as just a production factory for software.*

Processes might be considered as being separate from the organization, but cannot be seen as separate from the delivered services. In the examples given above, implementation of processes will, without doubt, lead to different implementations, with differing (sequences of) steps, management information, responsibilities, influence from and interfacing with customers and interfacing points.

### Managing ICT services will become a big problem

The complexity of managing a constellation (as described in figure 2) will be high. But nevertheless, this is what the market requires and the way in which market trends are developing.

There are two strategies which can be adopted to manage this complexity:

- Enlarge and make uniform the management process flow, by standardizing and add more governance.

This strategy has some severe disadvantages:

- Inflexibility towards suppliers. It will be difficult to change the supplier, because a new supplier has to adapt and implement the processes, which are dictated by an external force (customer).
- Loss of responsibility for supplier results. A supplier might deny responsibility, because regulations and processes on his internal process were designed by others.
- Organizations that deliver standard services or solutions for multiple customers may particularly disagree on this. Suppliers with multiple customers cannot and will not make their internal processes uniform for just for one customer. It would necessitate support for many different process implementations.
- A second approach is to reduce the need for management. This approach is particularly recognizable for application management.

### SOA AS A PARALLEL

Over several decades, when designing or maintaining applications, Application Management has dealt with a similar issue: how to manage the complexity of the information system or application.

The latest answer to this ongoing complexity is Service-Oriented Architecture (SOA) or Software as a Service (SaaS). This concept is the last step in a continuing line, which started with structured design (modularity), developed into object-orientation and finally transitioned to SOA.

The basic solution hides an increasing amount of knowledge about how things are built by defining the interfaces. With structured design, the structure and control flow within a program or function was hidden. With object-orientation the data was hidden and the only communication permitted was by procedures (methods).

SOA can now be seen as hiding even more, including the information model, the structure of the application and the underlying infrastructure and technology.

The only means of communication is by messages and calling services, which are fully independent from any technical implementation or solution. Everything has become a 'black box'. The same development can also be seen with shared service centers.

We think that the adoption of such an approach will be unavoidable for the world of ICT-management. This approach will lead to several concepts:

- *Suppliers are also becoming customers.* In order to deliver, they 'buy' underlying solutions from subcontractors. Sometimes they will act as a customer for a subcontractor. Alternatively, some suppliers provide standard solutions (from their customers perspective), but decide themselves as to the functionality of the solution, what the prices should be and what the kind of services of solutions they will provide.

1

2

3

4

5

6

7

8

9

10

11

- *Interfaces (in a broad sense) are required between the several suppliers and the demand organization.* Integrating and managing the ICT services will be done by designing, negotiating, contracting and managing the input- and output interfaces, as in Figure 3. The producing or consuming inside-processes will be a black box towards the outside world.
- *The integration question becomes dominant* (for most of the organizations). The question is how to integrate services of several suppliers and also who will be responsible for the integration. These questions will be dominant on both the demand side and the supplier side.
- *Processes will become an internal affair.* Customers are not interested in (internal) processes. The interfacing is becoming very important. The main questions are: “what do I need (to deliver)?”, “what am I doing myself?” and “how will this fit?”.
- *The future strategy of the supplier becomes a critical issue.* Questions will include: “what will be my role?”, “what will I supply?”, “what kind of services do I supply, and in which way?”. The need to compete with other suppliers has grown dramatically in the last decade.

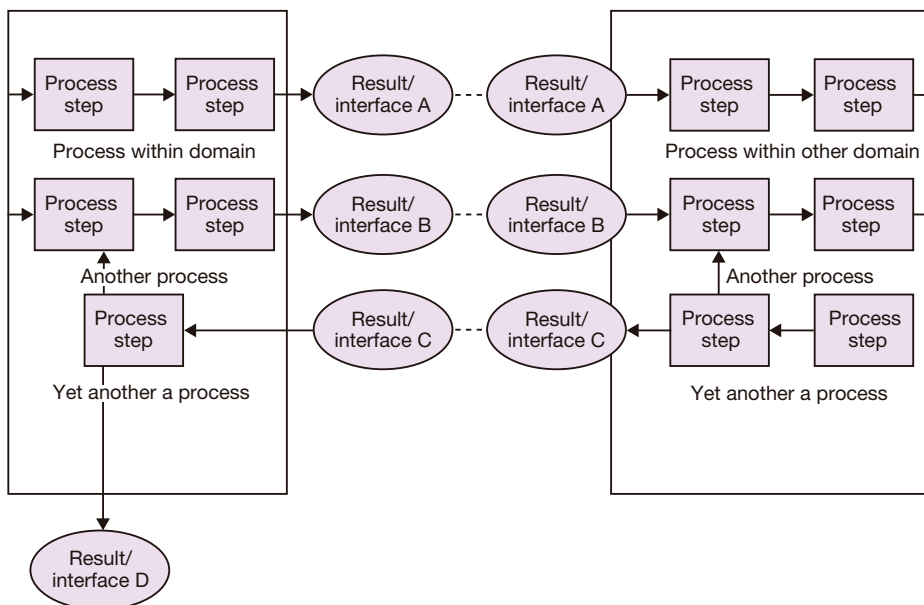


Figure 4 The role of interfaces

### Impact on process models

The conclusion is that it will be almost impossible to create integral service management processes. This causes inflexibility, high complexity and many problems when implementing. For a process framework such as ASL, this has many consequences:

- ASL will act as a component for application management, with the possibilities to 'integrate' underlying infrastructure management or other application management components. So it must be possible to act as an integration framework, but also as a 'stand-alone' application management component.
- ASL should provide the flexibility towards many forms of application management, and also support the differences for processes. Processes might be independent from the



implementation of an organization structure, but they are not independent of the delivered services. As an example: in bespoke applications the design of the application will follow the specification, created by the customer, but when building 'standard components' the design of the component might be the first thing to be considered.

- A basic concept of ASL, best practices as a starting point for the implementation of processes, will be more important. Best practices, which might be adapted to a typical situation, will make the implementation process manageable. Thus, the concept of a best practice has the same effect as a basic component in a SOA-architecture and implementation.

## IMPACT ON ASL

### Impact on the central concepts of ASL

The central concepts of ASL, as stated earlier in this article, remain valid. But the development as described above creates a frame of reference in which these themes of ASL are a logical consequence of the illustrated trend:

- The need for OCM and ACM are a logical result of this trend and are gaining more importance. The need to define and position your services in the future world will be critical, in order to survive in a world with many competitors. The same is true for the products, facilities or solutions, which are maintained.
- The distinction between external and internal quality becomes dominant. Services are purchased and contracted only on non functional requirements (external quality) and not on internal technical or technological issues. This also implies that managing the internal quality will be critical.
- The service team concept is a way of reducing the complexity. But we must also conclude that some organizations do not want to fulfill such a role, whilst some customers may not want to buy it. Many outsourcing contracts show this. Sometimes the service integrator will be business functionality.
- The use of a public domain standard will become normal. The question is not whether such a standard will be used, but which standard. Also, not which process model/framework will be important, but only the way in which you implement it. The processes will be an internal issue. Best practices will become a necessity, because the model might be the same, but the implementation will be increasingly different (taking account of the specific 'local' situation).

The impact of this trend also causes major changes in ASL. The most important changes will appear on the level of management processes, because the change is in how ICT-services are being bought, managed and combined.

The upper process clusters (ACM and OCM) will remain unchanged.

### The changes on management level

Therefore, the processes on the management level in ASL (the processes on the middle layer) are being changed rather significantly. One new process has been created and the content, underlying objects or names of the other processes on this level have been altered.

### Contract management

ASL version 1.0 had a process 'service level management'. The name of this process has been changed to 'contract management'. Not only has the name been changed, but also the objects, which are to be managed by this process. Issues which are to be managed, include:

1

2

3

4

5

6

7

8

9

10

11

- the responsibilities of supplier and demand organization
- the model of how and by whom services are being managed and funded
- the way in which demand and supplier co-operate and communicate
- the assumptions and conditions, prerequisite to the ICT-services
- the requirements on contract level and the translation to (functional) service levels

This means that the scope and the importance of the process have greatly widened.

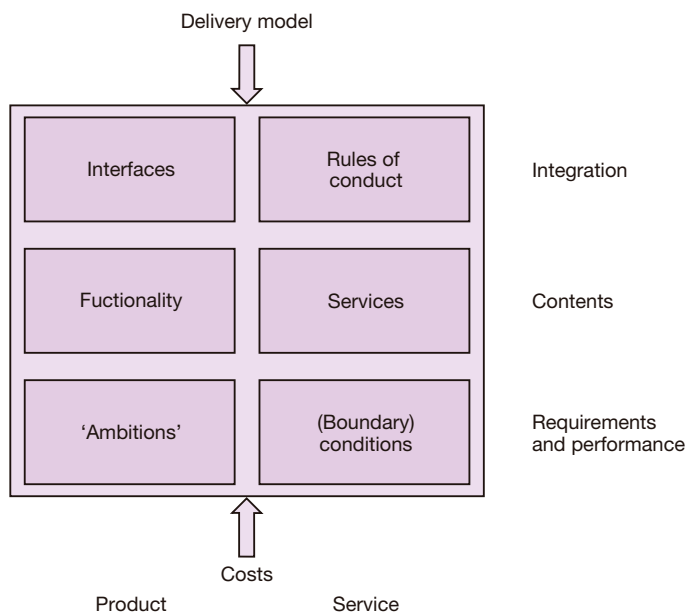


Figure 5 The agreements

### Subcontractor management

The described trend also leads to a situation, where the use of subcontractors will be the rule, instead of the exception. A big change in ASL is the introduction of a process 'subcontractor management'. Making the right contract and managing the (output) of the services, delivered by subcontractors, will be critical in the illustrated situation.

### Financial management

The name of cost management has been changed to financial management and the focus has been widened. This change also requires an explanation.

The original name in the ASL model was cost management, because the business case (from a point of view of the user organization) was positioned within Business Information Management (BiSL). Application management can not make a valid business case for a change in the business. They simply do not have the knowledge and they cannot be responsible for achieving the benefits. This assumption still holds.

Increasingly, application management will use cost calculation models, where application management does not charge the real costs (whatever they may be), but will charge by means of fixed prices.

This means a translation between external charges and internal costs. Therefore, this also requires an internal business case. The use and the role of subcontractors increases this need.

### Quality management

Quality management in ASL was responsible for the internal view on quality. This meant the quality of the product, the application organization, the quality system (the infrastructure for the application development and maintenance).

But some new dimensions are added:

- Quality management is responsible for ensuring that the internal quality of processes, products, organisations and quality infrastructure is sufficient to reach the contracted demands (external focus on quality).
- However, quality management is also responsible for ensuring that the solutions/ products, processes and results of subcontractors (in combination and integration with their own services and solutions) will be enough to reach the contracted demands. The completeness of own delivery and delivery of subcontractors will be the responsibility of quality management. Thus, quality management will be the central process on the management level of ASL.

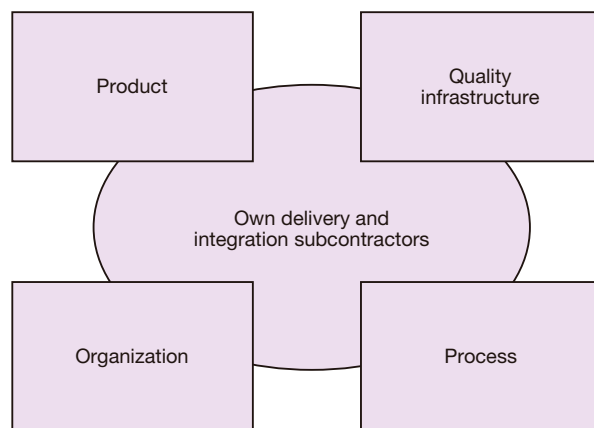


Figure 6 Objects within quality management

### Planning and control

The scope of planning and control also changes. The scope is not only the planning and control of time and capacity of manpower from the organization, but it also has to deal with the plan, check and act functions from subcontractor input. The focus and the goal does not change, the complexity is increased a little.

Another issue of increasing importance is that of managing the changes on the applications as a project, whilst still dealing with organizations departments.

### Changes on the operational level

On the operational level of ASL there are also some larger changes, but they are less fundamental than the ones on the management level of ASL. The most important changes are:

1

2

3

4

5

6

7

8

9

10

11

### 1. Configuration management

In ASL version 1 several best practices from the past could be found, such as naming conventions. In the new version this is removed. The relationship between the application and the underlying sources was not properly described.

The concept of services items and a service delivery database was introduced, and this is not heavily used in practice. The ASL review board has decided that this concept will stay within ASL. The underlying reason is the described trend towards shared solutions, where customers will have different versions of the object and will also buy different services. A logical consequence is that there will be an increase in the need for information about what versions are used and also about what the agreed services should be.

### 2. ICT Operations management

On an operational level the processes capacity management and availability management will be combined to the process 'ICT operations management'.

These processes have a similar flow, similar information and also impact on each other. Availability is increasingly harmed by a lack of capacity. The issue of capacity is still important, but because of the growing power of infrastructure it becomes less and less of an issue. In BiSL these processes (including continuity management) are combined. Continuity management in application management is still a separate process, with a different scope.

### 3. Changing names

The names of some processes are also being reconsidered. Discussions are still in progress as to the names of the processes 'software control and distribution', 'implementation' and 'incident management'.

The process 'incident management' will be renamed 'use support'. This name not only expresses a more proactive attitude towards users (most often business information management), but also the 'proactive communication' finds its place within this name.

The process 'implementation' will be renamed in version 1.1 to 'prepare and support implementation'. This name better expresses the activities within this process.

### 4. Changing names for used objects

Other changes will appear, but they will be minor and have less impact. Some of these changes are caused by the appearance of BiSL. For instance, BiSL uses the word 'specification' as an indication for the functionality requirements. The meaning of this word within ASL had to be changed in order to keep both models consistent.

Several remarks were also made with regard to the Dutch naming conventions for the strategic processes, but this will not have any impact on the English names.

### The next version of ASL

The next version of the ASL framework will adopt a structure that is relatively similar to the previous version. Most changes are to be found within the objects, which are managed within the processes, and in the ways how to structure these objects (see Figure 5).

Application Services Library, adapted to the IT-services world of the future

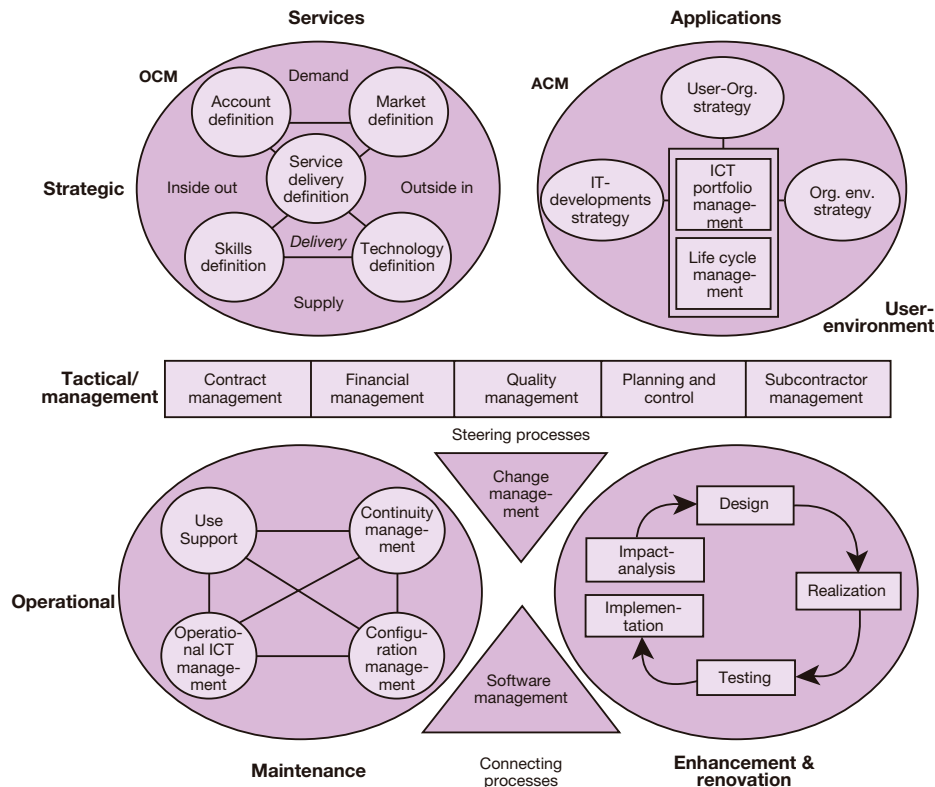


Figure 7 ASL version 1.1<sup>2</sup>

## CONCLUSIONS

### The biggest challenge: the identification and the integration of services

The market is growing towards a situation, where many ICT-organizations have to co-operate in order to deliver adequate and working ICT-services for customers.

One of the most frequently-asked questions is how ASL interfaces with other frameworks, such as ITIL or BiSL. In the process flows these connections were already mentioned, but they do not offer sufficient practical guidance.

The main message is that this integration is the biggest issue that needs to be solved. In this new world the customers will sometimes manage supply organizations and supply organizations will sometimes manage customers, with every type of variation in between. There will not be one standard solution.

Therefore, the question will not be how to integrate processes, but how to integrate services. The answer will be: by defining good and complete interfaces, and then by implementing processes to deliver and to manage the realization of the interfaces. Integration by interfacing will be the answer. This means that, for each required service, you will need good and complete interfaces in terms of services, costs of services, conditions, reliability, etc. It will be

<sup>2</sup> The names of some processes and clusters might still change

1

2

3

4

5

6

7

8

9

10

11

a critical factor when buying or designing these services. But it will also be an object which is part of the contract and contract management. This must be decided on each occasion.

Contract management and supplier management (within BiSL) will be the critical processes, just as contract management and subcontractor management will be within ASL.

#### **Standard process and standard integration will no longer be feasible**

Achieving management and control by implementing integral processes (for customer and supply organizations) is also an illusion. Integrating processes by using complete 'interfaces' will become the mechanism. Interfaces will provide the flexibility and simplicity that all parties require.

This means that a process cannot be implemented outside of the environmental context. It is too simple to expect ASL to provide a (mechanical) standard solution and standard answer to the integration and complexity needs.

Therefore, ASL can be seen as a standard solution, which has to be customized (implemented) to each situation for correct execution. It does not prescribe a single and uniform solution.

Meanwhile, ASL provides a framework and best practices which support this complexity (rather than opposing it). It provides best practices which will offer support when implementing the processes. It identifies the different situations.

#### **The role of process models**

As a consequence, there will be a change in the use and importance of processes and process models such as ASL and ITIL:

- Neither the process model nor the process are important as far as the outside world is concerned. A process will be an internal affair. As a consequence, choosing which model to use will be an internal decision.
- This does not mean that the use of processes and process models is not important: this will be the case more than ever. In order to 'survive' in the competitive world of IT services, delivering the contracted services is a critical factor, and processes will be an unavoidable means to achieve this.
- But each time the situation will be different and have different goals, services and results to be achieved. This means that a standard implementation will no longer exist; each implementation is new.

The use of best practices will significantly improve this, being used as the building blocks (components) as in SOA-architectures. Therefore, the collection of best practices will increasingly become a requisite.

We anticipate that the new version of ASL will be a step forward towards a more complex future. Best practices from other domains such as SOA have been adapted to give the answers, which are not solely to be found in our ICT sector, but which can be seen in all sectors.

**Remko van der Pols** (Netherlands) is a member of the architectural board of the ASL BiSL Foundation and is a Business unit Director for GetronicsPinkRocade.

*The author would like to acknowledge the contribution of the ASL Review Board, and especially of Mark Smalley.*

## REFERENCES

- Looijen, prof.dr.ir. Maarten van (1998). *Beheer van informatiesystemen*. Deventer (NL): Kluwer bedrijfsinformatie.
- Pols, Remko van der (2004). *ASL, a framework for Application Management*. Zaltbommel (NL): Van Haren Publishing.
- Pols, Remko van der, Yvette Backer (2006). *ASL, a management guide*. Zaltbommel (NL): Van Haren Publishing.
- Pols, Remko van der, Ralph Donatz, Frank van Outvorst (2005). *BiSL, a framework for Business Information Management*. Zaltbommel (NL): Van Haren Publishing, 2005
- Pols, Remko van der, Yvette Backer (2006). *BiSL, a management guide*. Zaltbommel (NL): Van Haren Publishing.

1

2

3

4

5

6

7

8

9

10

11

