

How to Support the Negotiation of Service Level Agreements (SLAs) for Your Client/Server Application?

Andreas Köppel
koepfel@cooperation-management.de

And

Dirk Böning
un7v@rz.uni-karlsruhe.de

And

Sebastian Abeck
abeck@cooperation-management.de

Institute for Telematics, University of Karlsruhe, Zirkel 2
76128 Karlsruhe, Germany

ABSTRACT

An increasing number of organisations decide to promote their core competence only. So, they outsource the required IT Services to an IT service provider and negotiate a Service Level Agreement (SLA) which is the contract between the provider and the organisation (customer). One of the major problems of SLA negotiation is the knowledge gap between provider and customer. The knowledge gap is the mutual lack of knowledge about organizational structures and technical aspects of the application environment. The proposed concept supports the standardisation of SLA negotiation for complex client/server applications. It is useful for service providers who have to establish SLAs repeatedly for the same application and for software manufacturers who want to support the usage of their product in outsourcing contexts. The concept is based on the application introduction process and therefore ensures completeness of the SLA by identification and decrease of the knowledge gap. It comprises several components that accompany the negotiation and allow the specification of the future SLA. The central component is the SLA Mapper that reveals the parts of SLA to be negotiated according to the nature and width of the knowledge gap. The concept and the components have been implemented for SAP R/3.

Keywords: Service Level Agreement, SLA, Application Management, Client/Server Application Introduction, Outsourcing.

1. INTRODUCTION

Our concept supports the standardization of the Service Level Agreement (SLA, [3]) negotiation for complex client/server applications. It is useful for service providers who have to establish SLAs repeatedly for the same application and for software manufacturers who want to support the usage of their product in outsourcing contexts. In consequence, software manufacturers may use this concept to offer a tool which

alleviates outsourcing their application and establishment of SLAs for their application. They could hand the tool to customers who decide to outsource and to providers who are interested in offering an outsourcing service for the application. The provider himself may use this concept to set up a standard process for SLA negotiation.

SLA Definitions

Service Level Agreements (SLA) are formal negotiated agreements that help to identify expectations, clarify responsibilities and facilitate communication between a service provider and its customer [8]. The provider may be internal (e.g. own computing center) or external (e.g. an ISP) to the customer's organisation. CCTA's [10] definition of SLA is: A written agreement or contract between users and the IT service provider which documents the agreed service levels for an IT service. Typically it will cover: service hours, service availability, user support levels, response times, restrictions and the service levels to be provided in a contingency. It may also include security and accounting procedures.

Dependent on the latter definitions, the SLA is a contract that should answer the following questions: Who delivers which service when? What happens if problems arise? What is the service and how is the service quality assessed? How to work changes into the SLA? These questions are not easy to answer when dealing with complex software applications like SAP R/3, Lotus Notes or any other Client/Server-based, enterprise-wide used application [5]. The common denominator for complex software applications is that their impact for the customer is great. The impact includes a great number of users (maybe several thousands), mission-critical functionality, 24-7 availability or the mapping of complex customer processes to the software functionality. In addition, these software systems cannot be introduced within days. It takes weeks or even months to plan, test and set up the system according to the required purpose.

Outsourcing Scenarios

The lifecycle of an application is roughly characterized by the introduction phase and the production phase. The introduction phase represents the complete setup of an application system for its required use. The production phase follows the introduction phase and represents the final usage of the application in the productive environment.

Three different outsourcing scenarios could be identified (s. Figure 1):

1. The providers accompanies the customer through the introduction of the application, i.e. the provider joins the customer at the beginning of the introduction.
2. The provider joins the customer during the introduction phase.
3. The provider joins the customer during the production phase (which includes the beginning of the production phase).

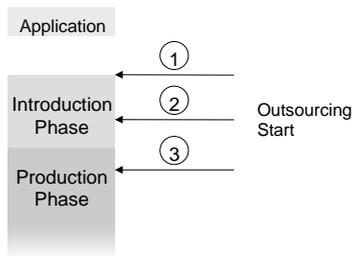


Figure 1: Outsourcing Scenarios

2. KNOWLEDGE GAP

SLA negotiation is characterized by the observation that customer and service provider are separated by a knowledge gap when commencing SLA negotiation. This knowledge gap comprises lack of experience, lack of knowledge about the organizational structure and lack of general knowledge with respect to the application, the application environment and the deployment of the application. I.e. knowledge in this context is threefold: general knowledge, deployment-dependent knowledge and organisation-dependent knowledge. General knowledge could be described as knowledge about the software which is obtainable from books, online help etc.. Deployment-dependent knowledge is gained by means of using, developing or testing the application. Organisation-dependent knowledge is about how the software fits into the enterprise structure. The knowledge gap is evident when looking at a customer and a provider at the beginning of the introduction of a new application software system: The customer has no explicit idea of the exact mapping of his business processes to the new system. This is, he lacks testing/usage experience and he cannot foresee how the application fits into the organizational structure of the enterprise. The provider will not be able to estimate the future complexity of the system at this early stage. He is neither familiar with the organizational structure of the enterprise nor with the customer processes. The maximum width of the knowledge gap is fixed since it comprises the entire amount of necessary information for successful SLA negotiation. The actual width of the gap depends on the individual and mutual knowledge of the negotiating parties.

Decrease of the knowledge gap enables customer and provider to successfully negotiate a reasonable and feasible SLA. The decrease of the knowledge gap cannot happen instantaneously. Tests cannot be executed at once. Experience with respect to the application deployment cannot be gained within days. Organizational structures cannot be set up in a moment.

Figure 2 illustrates the temporal constraints of decreasing the knowledge gap. In case the customer introduces the application system accompanied by the provider, the minimum time the customer needs to overcome the knowledge gap is depicted to the left. In case the provider joins the customer after the introduction (i.e. when the application system is already running in an productive environment) the minimum time the provider needs is shown to the right.

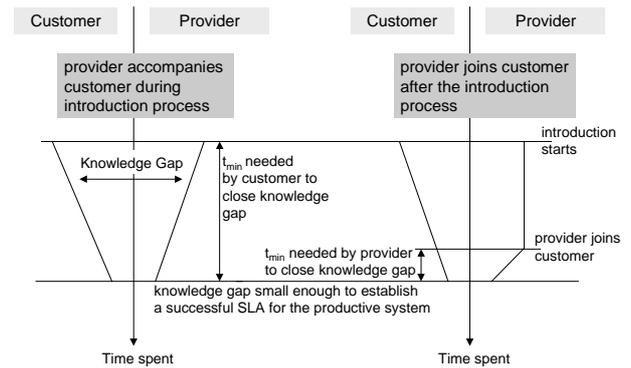


Figure 2: Temporal Aspects of the Knowledge Gap

SLA Negotiation Problems and Failure

As shown before, negotiation of SLAs has to respect the temporal constraints of knowledge acquisition.

So, what jeopardizes SLAs for complex software applications most? One of the major reasons is the knowledge gap between the customer and the provider. If both parties decide at a too early stage, i.e. the knowledge gap is too big to create a concrete and detailed SLA, the SLA must be changed subsequently, since the services agreed were not specified according to the actual requirements of the application environment.

Disappointment, delays and increase of costs must be expected. In the worst case the contract may be terminated - the SLA has failed [8]. In order to prevent SLAs from failing, provider and customer must identify their momentary knowledge gap based on a complete set of SLA relevant aspects. While closing the knowledge gap they negotiate more details and aspects of the SLA and finally obtain a reasonable, complete and feasible SLA.

3. SLA NEGOTIATION SUPPORT

Requirements

How to support SLA negotiation in order to prevent from problems and failures? In other words: How can the knowledge gap be identified and closed?

SLA negotiation support must fulfill the following requirements:

- Identification of the knowledge gap and the knowledge needed to close the gap

- Completeness with respect to all aspects of the future SLA and needed knowledge
- Reflection of application specific aspects
- Strategy to allow provision of the negotiation support

In order to ensure completeness and identification support it is necessary and appropriate to take into account the temporal aspects of knowledge acquisition in this context. I.e., the customer cannot gain experience needed at once. Therefore, closing the knowledge gap has time constraints that are best reflected by the step-by-step introduction of an application system. For almost every complex software a step-by-step introduction project is appropriate in order to get the system running in a productive environment and therefore it is assured that no aspect relevant to the application and its operation is missed. In most cases this project consists of the phases planning, requirements analysis and definition, implementation, testing and finally operation. In the case of SAP R/3 the introduction project is realized by ASAP (Accelerated SAP, [1]) that names the phases 'Project Preparation', 'Business Blueprint', 'Realization', 'Final Preparation' and 'Go Live & Support'. Other introduction strategies like DSDM (Dynamic Systems Development Method, [4]) are possible.

Thus, our SLA negotiation support is primarily based on the introduction project of complex client/server applications. On the one hand it is suited to determine the aspects of knowledge that are still missing when the customer and the provider start negotiation. On the other hand the introduction project is best suited to identify the nature and the essence of the knowledge gap in order to implement the components of the negotiation support.

Application of the Negotiation Support

The SLA negotiation support is to be applied basically during the specification phase of the SLA negotiation process.

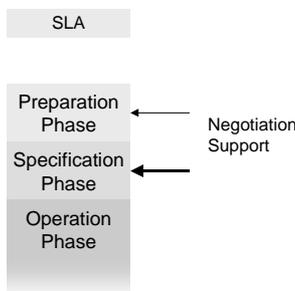


Figure 3: Application of the Negotiation Support

Preparation Phase: The provider and the customer agree on the principal services (as goal for the future SLA) that should be implemented. The provider calculates costs for different implementations of the services. These calculations enable the customer to initially estimate costs and thus improve the choice of possible services. The negotiation support helps to develop the goals of the future SLA.

Specification Phase: Based on the goals agreed during the preparation phase, provider and customer specify the SLA while implementing the services. The negotiation support is used at the beginning of the phase to determine the knowledge gap. Subsequently, the negotiation support

accompanies the specification phase and identifies the parts of the SLA to be specified or refined according to the width and nature of the knowledge gap. The end of this phase the SLA is complete and operational.

Operation Phase: The SLA is negotiated and applied. Further changes now depend on changing customer requirements. In general, these changes affect only parts of the SLA. If major changes are necessary provider and customer should re-begin the SLA negotiation with the preparation phase.

Investigation of provider-customer relationship showed [2] that the outsourcing project often starts at the end of the introduction phase or in the operation phase of the application. It is strongly recommended that the customer uses the negotiation support already during the introduction phase of his application in order to close his part of the knowledge gap as much as possible. Later, on starting the outsourcing project, these SLA relevant information could be applied smoothly to the specification phase of the SLA negotiation.

Components

In accordance to the above requirements and based on the application introduction, the key components for SLA negotiation support are the following:

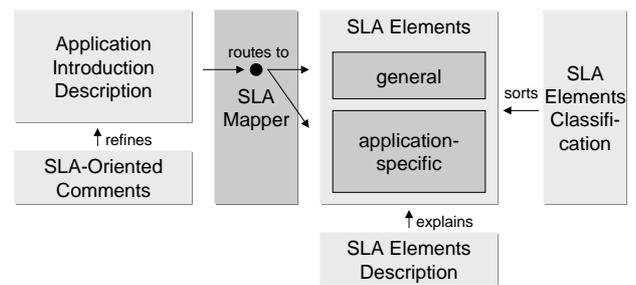


Figure 4: Negotiation Support Components

The components may be implemented as documents or software tools that give guidance on SLA negotiation for both customer and provider [7].

On the beginning of the SLA negotiation (SLA preparation phase and beginning of SLA specification phase) both parties will start working into the application introduction description and the SLA-oriented comments in order to determine their knowledge gap. While decreasing the knowledge gap, they apply the mapper and define and refine the SLA elements. The complete set of defined and refined SAL Elements represents the entire SLA. The SLA element description and classification help to understand the nature, importance and structure of the elements.

Application Introduction Description: The application introduction description is normally provided by the software manufacturer or anyone with sufficient knowledge of the introduction. It outlines the introduction phases, sub-phases and states of the introduction and thus, gives an overview on the knowledge that is needed to overcome the knowledge gap between provider and customer. This description is not yet SLA-specific. The structure of the

description depends on the introduction process of the application. It reflects introduction phases and states as sections and subsections. The complexity of the structure depends on the introduction process and therefore on the complexity of the application system. E.g. the standard SAP R/3 introduction process ASAP results in a hierarchical and sequential structure with 5 major sections and 3 subsection levels.

SLA-Oriented Comments: The SLA-oriented comments evaluate the application introduction with respect to the future SLA. The comments take into account currently available knowledge and give further informal information on SLA aspects of the corresponding part of the introduction description. In addition, examples and experience reports may be provided. The comments should allow determination of the depth of SLA detail. After each phase of the introduction the knowledge gap is getting smaller and more details of the SLA can be specified.

SLA Elements: The SLA elements can be divided into two groups: general and application-specific elements. SLA elements should not be confused with service elements. SLA elements may represent service elements.

The general elements should be part of every SLA and do not depend on the application. They will be provided with the concept. Among others general elements are mutual contact personnel, change process of the SLA, periodicity of reviews, costs, duration of the contract, sanctions in case of service level violation and obligations of the provider in case of contract termination.

Since each application has its own characteristics, elements that are specific for the actual application should be defined in the SLA. Application-specific SLA elements are provided with an implemented negotiation support and should represent the maximum set of application-specific elements. I.e. all possible elements should be named. Of course, this requirement is difficult to achieve, but it encourages the organisation that implements the negotiation support to strive for completeness. Application-specific elements comprise service descriptive elements, service reporting elements, etc..

SLA Mapper: The SLA Mapper maps states and phases of the application introduction to the SLA elements and their possible contents. As the SLA-oriented comments specify the depth of detail of the SLA for a section or subsection of the description, the mapper specifies the involved elements with the section. Thus, the mapper routes to the knowledge-adequate subset of specifiable SLA elements. The elements routed to can either be defined or refined. Element definition means that the element is specified the first time during negotiation. Refinement means that the element has already been defined, but the level of detail of the element specification is increased.

Since, the mapper should refer to all SLA elements, the complexity of the mapper is dependent on the number of SLA elements. But it depends not on the detail of the application description. Although, the number of sections and subsection levels represent a measure for the complexity of the description, the number of links between the mapper and the

description is arbitrary. It is valid to use one link for the union of several sections or subsections.

SLA Element Description: A description for each element of the SLA is attached. The description will help both parties to get clear and unique understanding of the meaning and importance of the element. Furthermore, the element description may point out possible conflicts arising from diverging interests of provider and customer. Usage of standards like the IT Infrastructure Library of the CCTA [6] may be deployed.

SLA Element Classification: Finally, a classification should help to get better overview of the different elements and alleviate the negotiation of the SLA. The classification may comprise categories like overall importance for the SLA, delegatability (the SLA element may be delegated completely to either of the parties), measurability (the applications is instrumented in a way that allows the measurement of the this element) or dependence on other SLA elements.

Component Provision: The general structure of the negotiation support components except the application introduction description is application-independent. It is provided with this concept. The content of the components except the content of the general SLA elements is application-specific and must be developed independently from the concept. Only the general SLA elements are supplied with the concept. In order to implement the application-specific components of the SLA negotiation support and application-specific content of the support components the following steps have to be carried out:

1. Specification and description of the application introduction (by the software manufacturer or the service provider)
2. Analysis of the introduction according to the gained knowledge and implementation of the SLA-oriented comments.
3. Together with the application-dependent introduction description, the SLA-oriented comments, the application-specific SLA elements can be identified Then the SLA element description and the element classification can be implemented. Additionally, measurement and administration tools provided with the software, could be analyzed in order to identify further SLA elements.
4. Implementation of the SLA mapper that describes the relationship between the SLA elements and the introduction description (= the knowledge of both parties according to the application introduction).

4. EXAMPLE SAP R/3

Subsequently, the main aspects and advantages of the negotiation support concept are demonstrated with the client/server software SAP R/3. The following sections also provide the description on how SAP R/3 can be fit into the negotiation support concept.

Why SAP R/3?

SAP R/3 is a good example to verify the concept of SLA negotiation support. R/3 is a very complex client/server software. Therefore, both the provider and the customer have to overcome the complexity of R/3 in order to negotiate an effective SLA. So, we believe that the implementation of the concept for R/3 will lead to more effective SLAs and more satisfaction for both parties.

The SAP R/3 landscape is explained briefly in order to illustrate its structure and complexity. The landscape is built of three R/3 systems (s. Figure 5). These systems are linked together in order to ensure that development or customization of arbitrary nature is thoroughly tested before the changes are applied to the production system. Only the production system holds all unique business data entered by R/3 users.

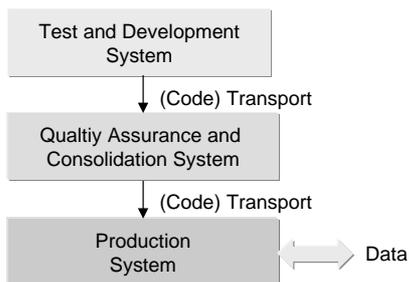


Figure 5: R/3 Landscape

Each system consists of a database and up to tens of servers. The user is connected to the system via WWW or the R/3 client software (SAPGUI). The servers exchange data with other R/3 systems and with other applications by means of standard and custom interfaces.

The next paragraph describes the knowledge gap between provider and customer on outsourcing SAP R/3. Section 5 will cover an application introduction description for R/3 according to AcceleratedSAP (ASAP). Section 6 outlines the general and R/3-specific SLA elements. Section 7 part illustrates briefly the SLA Mapper. Finally, section 8 explains the remaining components.

Knowledge Gap

SAP R/3 is a very complex application system and can be configured manifold. Even if R/3 is used in standard business processes many configurations are possible. Inasmuch, the configuration desired by the customer cannot be totally derived from the experience of the provider. If the introduction process of the R/3 system is not yet begun, both parties have a big knowledge gap. The customer may define his requirements by experience, but still some aspects of the future SLA are undeterminable at the beginning.

The definition of the SLA Element backup strategy is a good example to explain the knowledge gap in the R/3 context: The customer specifies that he will need one backup a week for a maximum of 10 Gigabytes of data. The provider will calculate the price for this backup service. Both parties agree on this service level and the service and the service level is documented in the SLA. Then, the provider may already buy and/or setup the tape devices. If this documented backup service level is considered as final, i.e. it will not be adjusted during the introduction process of R/3, problems arise most

likely. Since, the backup strategy will be tested in a real system environment during the introduction project with the actual amount of data of the system, the customer might need more backup capacity as expected. Unfortunately, provider and customer in the example have already fixed the maximum data amount. So, the newly gained knowledge obliges changes to the SLA. Futile effort has been invested as the backup service level has been fixed at a too early stage.

The given example illustrates one of the main reasons why the final specification of the SLA elements cannot be made with a big knowledge gap and that element specifications have to be refined at the time when both parties gain adequate knowledge.

5. APPLICATION INTRODUCTION DESCRIPTION

Accelerated SAP (ASAP [1]) is a set of tools and a project guide for an efficient R/3 introduction. This introduction project is designed to implement R/3 within 6 to 8 months. It consists of five consecutive phases: Project Preparation, Business Blueprint, Realization, Final Preparation and Go Live & Support. In the following you can find a description of each phase.

Phase 1: Project Preparation

During this phase all planning with respect to the project course is executed. The definition of the project objectives is included which helps to assure that all project participants share the same goals.

Very important for the future need of resources is the introduction strategy. It defines whether to go live (i.e. using R/3 productively) with all business functions of the system, also known as the 'big bang'-strategy or to go live with only a subset of the required business functions first. If the 'big bang'-strategy is used all resources have to be provided at once, which will be more expensive and potentially risky.

The project plan defines all tasks and processes during the implementation of the system. Each task or process will be linked to a role of the introduction project team. To get a better control over the progress of the project, milestones will be defined. Additionally, an initial specification of the system environment (needed systems, client-, release- and transport-strategy) is elaborated. It is called the 'Initial Sizing'. From a technical perspective, all technical requirements must be defined.

Phase 2: Business Blueprint

In this phase the development system is used to plan how to transfer the business processes into the R/3 environment. The customer defines the system and network infrastructure and the development administration tasks.

According to the business processes that have to be realized in the R/3 system the need for interfaces to other systems or special data transfer procedures is identified. If the functionality of the standard R/3 system is not sufficient, modifications that extend the system have to be defined.

The result of the Business Blueprint phase is a plan that describes the realization of the existing business processes in the R/3 environment.

Phase 3: Realization

Now, the requirements of the Business blueprints will be implemented in the development system and transferred to the consolidation system. The latter system is used as simulated productive system. The systems will be configured such that the business processes can be executed. This is called customization. Aside from the configuration and customization settings the data transfer programs, interface programs, system modifications, reports and forms must be realized. Additionally, the authorization concept and archiving strategy is implemented. Finally, all configuration settings, programs and strategies are tested in a final integration test. The integration test must be planned carefully.

Phase 4: Final Preparation

The Final Preparation phase comprises user training, cutover, support structure planning and preparation of the production system. Cutover is the transfer of business data into the production system. The support structure requires definition of problem priorities and time ranges for error resolution. A help desk has to be planned and installed. The preparation of the production system implies system tests and implementation of the production system administration.

Phase 5: Go Live & Support

This phase marks the end of the introduction project. The cutover will be executed and the production system will go live. The support structure must be checked and modified such that it will be effective on a long-term basis. The long-term tasks of the system administration should be defined and processes for the R/3 change management and release changes must be prepared. Finally the project should be reviewed and evaluated if all the goals and milestones were successfully reached.

6. SLA ELEMENTS

Neither the following collection of general SLA elements nor the collection of the R/3-specific SLA elements is complete, since completeness is beyond the scope of the paper.

General SLA Elements

General elements mainly deal with contractual aspects of the SLA, since the contract conditions should be specified in the SLA. The following elements can be counted as contract conditions: period, term of notice, extension possibility, starting time, expandabilities, periodic review, change log and change management of the SLA. Other contractual elements are basic costs and the costs for options and extensions. Also the sanctions for service level violation and obligations of the provider in case of contract termination should be defined.

Non-contractual general SLA elements are the intention and purpose of the SLA. Both help the SLA negotiating parties to understand that the SLA should not be used to put pressure on the other party.

Contact persons who are responsible for SLA negotiations and changes should be defined and definitions should be documented to make sure that each party has the same understanding of terms. Documentation aspects, privacy and the SLA change process belong also to the general SLA elements.

R/3-Specific Elements

The R/3-specific elements mainly refer to technical aspects of the application. The definition of system environment comprises the number of systems, release version, databases, operating system, frontends, R/3 modules, printers, other peripheral devices and the network infrastructure. In order to ensure that the configuration of the customer is considered, modifications, reports, forms, interfaces, data transfer, backup and recovery strategies, archiving, test and authorization strategy should be specified. If required, a description of the planned cutover helps both parties to see the risks of the introduction process. Operational aspects like system administration, code transport and R/3 change management have to be considered in order to assure that all operational tasks will be fulfilled on a long-term basis. Security issues and connection to the WWW are also part of the SLA.

Each system of the R/3 landscape has its own requirements in terms of performance. Therefore, it is necessary to define the client strategy (client is used in the R/3-specific way as independent business unit from the view of an R/3 system) and the number of users for each system. Also the availability can differ from the development to the production system. Availability and response time are very important elements for the production system. They refer also to planned downtime and measurability or at least strategies for the measurement of these technical parameters.

Definition of the support structure enables both parties to establish effective communication in case of problems. The support structure may also include communication to third-party organisation like the SAP support or sub-contractors. Customer and provider have to agree on procedures that handle trouble tickets in terms of priority level, measure and problem solving.

7. SLA MAPPER

The main task of the SLA Mapper is to create a link between the SLA elements and the knowledge of provider and customer at the end of each phase of the introduction process. In the following, some examples are presented that explain how SLA elements are specified after the phases of the ASAP introduction.

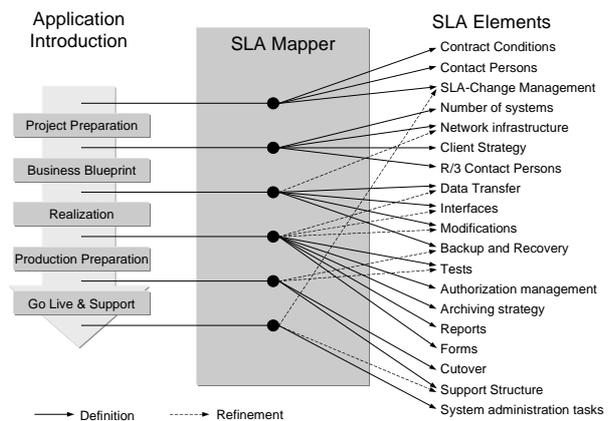


Figure 6: R/3 SLA Mapper

Before Phase 1: Project Preparation

Before the Project Preparation both parties agree on an outsourcing relationship. Now they are able to negotiate the general SLA elements. E.g. the general contact persons, who are responsible for changes on the SLA can be defined. The contract conditions like the period, term of notice, extension possibility, starting time, expandabilities, periodic review, change log and change management of the SLA can also be specified before the beginning of the application introduction process.

After Phase 1: Project Preparation

After the project preparation the customer has defined the number of systems. Thus, the SLA element 'number of systems' can be transferred into the SLA. The customer will also set up the requirements for the network infrastructure. Its description will not be very detailed, but it should already appear in the SLA. The customer has specified a client strategy for each system and has chosen an introduction strategy. Now, availability of the systems with respect to the 'when' and the 'in which scope' can be specified in the SLA such that the provider is able plan the needed resources.

Besides the technical aspect, both parties have to name their R/3-specific contact persons. The names and roles of these contact persons should be written down in the SLA to guarantee an effective communication between provider and customer.

After Phase 2: Business Blueprint

After the creation of the business blueprint the customer has the knowledge of the nature of legacy data transfer. The data transfer element in the SLA can be specified. In order to connect R/3 to other systems within the company the customer has to specify the interfaces needed. If the standard functionality of the R/3 system is not sufficient, the customer will need system modifications. These modifications must appear in the SLA.

For data protection aspects the backup and recovery strategy is defined in the business blueprint and can be taken over into the SLA. After the project preparation the network infrastructure was specified in the SLA. The business blueprint allows to plan the infrastructure more detailed. So, this element is refined.

After Phase 3: Realization

During the Realization phase the business blueprint is implemented and tested. The important system tests should be defined in the SLA, so that both parties participate in these tests and obtain their particular results. The authorization management of the future productive system has been specified and tested. So, it can also be fixed in the SLA. The archiving strategy of the customer is also defined in the SLA. The need of R/3 reports and forms can also be described in the SLA. The realization tests allow for modifications, interfaces and data transfer described more detailed and concluded.

After Phase 4: Production Preparation

After the production preparation the cutover to the final productive system is planned. The entire cutover is very

complex and comprises transfer of all data and customizing settings to the production system. So, it has to be specified in the SLA.

Another important aspect is the support structure. Each party has to install a help desk. Relevant aspects like communication to and between the help desks, support levels, availability of the support is also part of the SLA.

In addition to the system tests of the previous phase, additional tests referring to the backup and recovery return results the refine the corresponding SLA element.

After Phase 5: Go Live & Support

After the go live the support structure can be refined to its final state. Also the final system administration tasks of the provider will be written down in the SLA. In principal, this phase has no end, since system requirements or outsourcing requirements will change after the go live. The nature of the changes and the change management process are different from changes of the past phases. Therefore, both parties have to adjust the change management element for its use in the production phase of the application.

General comments (SLA Mapper)

The explanation of the R/3 mapper has shown its general structure. This structure corresponds to the scenario where the provider accompanies the customer's R/3 introduction project. If this is not the case and the customer will outsource the system during or after the introduction project, the mapper is still applicable without changes of the structure. Provider and customer work through the application introduction description and analyze their knowledge gap. If the R/3 system is already productive, they begin at phase 5 of ASAP and they specify all elements the SLA mapper has defined and refined in the previous phases. Thus, the mapper is valid either if customer and provider step through the introduction process together or if the customer is already productive with his system.

8. OTHER COMPONENTS

SLA oriented comments

It is beyond the scope of the paper to provide an example since the comments refer to ASAP details that were not introduced to the reader. See [2] for details.

SLA Element Description

In order to illustrate the nature of the SLA element description, one example from [10] is provided: Availability = The ability of a component or IT service to perform its required function at a stated instant or over a stated period of time.

SLA Element Classification

The classification of a the SLA elements improves the readability and effectiveness of the SLA negotiation. An example for a classification class is the measurability of the element in CCMS (Computing Center Management System, the R/3 internal management tool). An element like response time could be classified as measurable. Another classification class is the delegatability. SLA elements as backup and recovery could be delegated completely to the provider. Backup related tasks need not to shared between the provider and the customer. Then, if no task sharing is agreed, it is

completely the provider's task to successfully provide the backup service. No further co-operation of the customer is needed.

9. CONCLUSION

The proposed concept of SLA negotiation support offers a wide range of application scenarios reaching from informal informational use to application introduction accompanying SLA negotiation. It offers a clear structure and a well defined approach to SLA negotiation. Decomposition of the negotiation support problem into different components allows for decomposition of the implementation as well as a clear structure of the implementation. The components even permit selective implementation and selective use.

In future, more detailed usage instructions in different outsourcing scenarios will be provided. These enhancements augment the simplicity and ease of use.

The SAP R/3 implementation of the concept will be refined by means of interviews with customers and providers. These interviews adjust the focus on the most important aspects of the R/3 SLA negotiation support and help to further validate the concept. Hopefully, a complete sample SLA will be provided.

10. ACKNOWLEDGEMENTS

SAP AG, Germany.

Nicolai Jordt and Rudi Marquet at SAP, Walldorf, Germany.

11. REFERENCES

- [1] SAP AG, ASAP for 3.1, Walldorf, Germany, 1998.
- [2] D. Böning, Service Level Agreement Negotiation Support demonstrated for SAP R/3, Diploma Thesis, University of Karlsruhe, 1999.
- [3] A. Caine, Negotiating an Effective Service Level Agreement - I & II, Gilbert & Tobin, <http://www.gtlaw.com.au/gt/pubs/negotiating.html>, 1997.
- [4] DSDM Consortium, Dynamic Systems Development Method, Version 3, Farnham, UK, 1997.
- [5] H.-G. Hegering, S. Abeck, and B. Neumair, Integrated Management of Networked Systems -- Concepts, Architectures and their Operational Application, Morgan Kaufmann Publishers, 1999.
- [6] Central Computer and Telecommunications Agency (CCTA), The IT Infrastructure Library - An Introduction, 1994.
- [7] International Telecommunication Union (ITU), TMN Management Functions, M.3400, 1997
- [8] N. Karten, How to Establish Service Level Agreements, Karten Associates, Randolph, MA USA, 1998.
- [9] H. Kendall, Service Level Agreements, The Help Desk User Group, Orpington, UK, 1996.
- [10] Central Computer and Telecommunications Agency (CCTA), Quality Management for IT Services - IT Infrastructure Library, 1994.