RENARDUS: PROJECT DELIVERABLE

Project Number:	IST-1999-10562
Project Title:	Reynard – Academic Subject Gateway Service Europe
Deliverable Type:	Restricted

Deliverable Number:	D1.2
Contractual Date of Delivery:	31.3.2000
Actual Date of Delivery:	30.3.2000
Title of Deliverable:	User Requirements for the Broker System
Work package contributing to the Deliverable:	WP1
Nature of the Deliverable:	Report (internal deliverable)
URL:	Http://www.renardus.org/deliverables/
Authors:	Kimmo Tuominen, Janne Kanner, Manne Miettinen, CSC Rachel Heery, UKOLN (chapter 2)
Contact Details:	Kimmo Tuominen, CSC – Scientific Computing Ltd., P.O. Box 405, FIN-02101 Espoo, Finland, email: ktuomine@csc.fi, phone: +358 9 457 2078, fax: +358 9 457 2302, URL: http://www.csc.fi

Abstract	Functional requirements of the Renardus broker system are analysed from service provider and end-user perspectives. The service provider requirements are collected from a survey answered by Renardus participants. The viewpoint of end-users is represented by use case scenarios and collected results of end-user surveys. The results show that Renardus should be based on a distributed architecture and it should support both searching and browsing.
Keywords	Functional requirements, broker system, user needs, information gateways, digital libraries

Distribution List:	Renardus participants, Ian Pigott (project officer, EC)
Issue:	1.0
Reference:	
Total Number of Pages:	37

TABLE OF CONTENTS

PART II - MANAGEMENT OVERVIEW

DOCUMENT CONTROL

Issue	*Date of Issue*	*Comments*
V0.1	9.3.2000	First draft of final deliverable based on template.dot.
V0.2	15.3.2000	Second draft with conclusions.
V0.3	28.3.2000	Third draft with chapter 2 included.
V1.0	31.3.2000	The published version.

EXECUTIVE SUMMARY

User requirements for the Renardus broker system were collected in two parts. Firstly, service providers filled in a questionnaire concerning their requirements for the broker functionality. Secondly, use case scenarios were constructed from the end-user perspective. These scenarios were partly based on the results of the questionnaire for the service providers. The use case analysis work will continue after the formal publication of this deliverable. It is a way to carry further the analysis being presented here by concretising the user requirements. In the conclusions also the results of deliverable 6.6, in which the end-user needs based on existing user surveys are analysed, were taken into account.

The service provider's requirements were divided into five categories in the concluding chapter. The first three categories 'essential requirements' 'preferable requirements' and 'desirable requirements' follow the proposed definition given in the task plan of WP1. The two other categories emerged from the answers. They were 'functionality not requested' and 'problematic areas requiring discussion'. Essential requirements are defined in the document as the kind of features that must be implemented in the Renardus pilot system. Preferable requirements mean features that may be implemented optionally in the pilot and ought to be implemented in the final broker. Desirable requirements may be considered in the future but not in the pilot. The main conclusions to be drawn from the answers are described below.

The first conclusion concerns the architectural model. The respondents favoured a distributed model over a centralised solution. There should be no central repository to which all metadata is routinely copied, only a centralised subject index to forward the queries to relevant gateways. Another option here would be to implement a centralised index with decentralised databases. Regardless of the way the index is implemented, it should be updated quite frequently, preferably daily.

Renardus should support Dublin Core semantics and RDF/XML syntax for metadata records. Z39.50 and WHOIS++ should be used as protocols for communication between Renardus and the brokered gateways.

Both searching and browsing should be supported in the Renardus broker. Search functionality must be quite fast and it should be possible to restrict the search by using separate metadata fields. The end-users should also be able to use properties and origin of metadata (metadata creating date, etc.), language of the resource and location (country) of the resource as search criteria. There is also a need for mappings between metadata formats to support at least a consistent presentation of search results. Mappings between classification schemes are also needed so that the cross-browsing functionality can be implemented in the pilot system.

The answers show a strong consensus for transparency of the system. Thus, users should always be aware which individual gateways provided the search results. In addition, gateways should be given the possibility to construct their own user interfaces to Renardus or at least to make some kind of customisations to the existing user interface (i.e., by translating the English user interface to the native languages of their own user communities).

Renardus should also be able to collect statistics of the usage of the system. The exact content of these statistics remains to be discussed.

In the future Renardus should also be able to merge duplicate metadata records intelligently, support sophisticated cross browsing and be able to check the validity of links to resources. However, these features should be implemented in the eventual system, not in the pilot version of it.

It became also dear that following functionalities are not desirable in the Renardus context: 1) one large metadata repository for sharing metadata, 2) automatic translation of free text fields in metadata records or whole resource documents, 3) tools for the end-users to convert resources on the fly to another format or to a packed form, 4) automatic full-text indexing, 5) user authentication, 6) access control to resources, 7) automatic tools for classification of resources and 8) automatic resource checking tools that warn if metadata is out of date. The respondents also felt that the Renardus system should not require national databases or brokers in each country that takes part in the project. In addition, brokering to national bibliographies, other references databases or to digital collections of archives and museums was not seen as a necessary feature of the Renardus broker.

End-user requirements for the broker system concern search functionality and usability issues. End-users come mainly from the higher research and education community. The broker system will be used both via LANs and via modems. End-users prefer searching to browsing and do normally quite simple searches. End-users need an intuitive and simple-to-use system of quality-controlled information resources.

There are also some issues that seem to need further discussion between the project partners. These issues concern, for example, the architecture of the system, the type of statistics collected by the system and the way metadata is possibly shared between service providers.

SCOPE STATEMENT

The purpose of this document is to explore the functionality of Renardus from the perspectives of end-users and service providers. It should inform decisions regarding architectural design and set priorities for functionality. Along with deliverable 1.1 (Evaluation Report of Existing Broker Models in Related Projects) it will provide the basis for deliverable 1.3. (Specification of Functional Requirements for the Broker System) that will be ready in May 2000. The analysis being performed in the current document will also feed into the work of WP6, as a basis for development of the data model for the Renardus broker.

PART III - DELIVERABLE CONTENT

INTRODUCTION

The purpose of this document is to explore the functionalities of Renardus from the perspectives of end-users and service providers. The functional requirements are formulated on the basis of the analyses of service provider and end-user functional needs. These requirements should inform decisions regarding architectural design and set priorities for functionality. They also affect the design of the data model of the Renardus broker.

According to the technical annex of the Renardus project [1, pp. 17-18] and Renardus WP1 task plan [2] the tasks of deliverable 1.2 are:

? ? To make high level user requirements specification where users are considered as both 'end-users' and as 'service providers'

? ? To identify the users of the Renardus service and how they relate to the overall architectural framework

? ? To examine different models for delivery of the broker service to users

? ? To collect requirements of gateway providers as to the way in which their data collections are brokered to

? ? To formulate general brokering requirements concerning access (transparency, speed), synchronisation of updates, maintainability, dependency on gateways, efficiency (efficient use of network capacity), etc.

? ? To form the basis for the functional specifications and modelling

Furthermore, the task plan [2, p. 3] suggests that:

> *[F]unctionality is divided into 'essential', 'preferable', and 'desirable'. Essential functionality must be implemented. Most preferable functionality should be implemented but there may be some trade-offs. Desirable functionality will only be implemented if it proves straightforward, or if there is sufficient effort left after essential and preferable functionality has been implemented.*

The present document is structured as follows: Firstly, the analysis of the questionnaire (see annex A) sent to Renardus participants is presented (chapter 1). The questionnaire was aimed at mapping the service provider requirements for Renardus. Results of the analysis of the questionnaire inform preliminary use case scenarios from the end-user perspective that are presented in chapter 2. The work with use case scenarios will continue in the Renardus project after the formal publication of the present deliverable. It is important to describe step-by-step the interaction between the actor and the system for achieving some of the required goals. Chapter 3 is the concluding chapter in which general user requirements for Renardus are presented.

The general requirements formulated on the basis of functional needs of the end-users and service providers should be reflected, for example, in the architectural model chosen for the pilot system. However, one should keep in mind that because of the preliminary stage of the project, all the needs and requirements are perhaps not crystallised. We suppose that both end-users and service providers do not exactly know what they want or need at this stage. Thus, the results presented in this document should be viewed as a snapshot of the situation taken at the beginning of the project. If we had a follow up survey in 2002, the answers might be very different. Nevertheless, even if some of the areas covered in this document require re-visiting over the life of the Renardus project, main decisions concerning broker architecture and functional requirements have to be made before WP2 starts to implement the pilot software for Renardus.

Kimmo Tuominen, Janne Kanner and Manne Miettinen from CSC are responsible for the content of the document, except for the chapter 2 about use case scenarios, which is written by Rachel Heery from UKOLN.

We would like to thank Leif Laaksonen and Markus Sadeniemi from CSC for comments and support during the preparation of the document. Main results of the questionnaire for service providers were also presented in the Renardus technical meeting in Denmark at the end of February. In this meeting we received some valuable comments and criticism for which we would like to express our gratitude.

1 SERVICE PROVIDER REQUIREMENTS FOR THE BROKER SYSTEM

This chapter is based on a survey of service provider requirements for Renardus (see annex A). The survey was sent to all the participating organisations of Renardus on 11th of February 2000. Eleven of the twelve member organisations answered in reasonable time limits. The answers from these organisations are collected to annexes B (answers to closed questions) and C (answers to open questions).

The questionnaire comprises mostly of statements for which the respondents had to choose one option from the scale of 5 to 1 (5=essential, 4=preferable, 3=desirable, 2=not necessary, 1=should never be implemented or 5=strongly in favour, 1=strongly against). There were also a few open questions after each theme and a space for comments at the end of the questionnaire. Each participant organisation of the Renardus project completed only one form. Thus, the personnel in these organisations had to agree on the content of the answers.

At this phase of the project, it is necessary to analyse the functional needs concerning the Renardus system of actual project participants. However, it became evident that some of the participants already filled out the form in liaison with other service providers in their country. Because the time span to the deadline of the questionnaire was so short (only 10 days), we could not ask all the participants to discuss extensively about the questionnaire in their countries.

In the following, we present an analysis of the collected data. At the points in which this seems relevant we refer to a question by giving its ID in brackets after a sentence or chapter analysing the content of the answers to it. Occasionally, we also present the average score of a certain question or cite the answer to an open question from some respondent. However, as all the data is presented to the reader, there is no need to be so explicit in the analysis as otherwise would have been necessary.

1.1 Architecture

Majority of the respondents favoured distributed architecture. Although there were two respondents supporting a purely centralised solution, the opposition against one European master database was evident. However, query routing based on a centralised index got fairly strong support. Query routing provides the possibility to save both time and network bandwidth by forwarding queries only to gateways that most probably will contain relevant information. This kind of functionality can and should be implemented in distributed architecture models.

As remarked by Dovey [3], centralised index could mean two things: 1) manually or automatically created collection level metadata or 2) centralised index with decentralised databases of records. In the first case, search query will be performed first against the central index and after that it will also be performed in those database servers to whom the query is routed. In the second case, all the index data exists on one server and thus the search is performed only there. When the broker displays the search result list, the actual record can be obtained directly from the information gateway. It is also possible that the user is given an URI of the record in a certain information gateway or gateways when search results are presented. Thus, the user goes to an actual information gateway server only to see the whole record. Both of the possibilities 1 and 2 can be considered when implementing the Renardus system. However, the latter would probably make the search functionalities of Renardus much faster.

If Renardus is based on a completely distributed architecture the open question is how many information gateways can be reliably cross-searched (cf. [3]). The actual limit here is of course related to the scalability of the search technology implemented. However, too long searching times would make the Renardus system quite useless to ordinary end-users.

Quite surprisingly, the attitudes of respondents towards a common repository for sharing metadata were not very favourable. The manual cataloguing of metadata is a time consuming and expensive job. Evidently some of the respondents fear that they would loose more than gain by giving the fruits of their work for free to other organisations. However, one should keep in mind that experimenting with metadata sharing is one of the tasks

set for the Renardus project as a whole. Thus, there might arise a need in the future for a smaller centralised database or databases for importing and exporting metadata between information gateways.

It is also possible to enable metadata sharing in a completely distributed environment by building an own user interface for that purpose. The interface would present search results as whole metadata records using a common syntax and semantics. It would also be possible to import the found records into one's own system. The use of the interface should be restricted so that only the service providers agreeing to recycle their metadata had access to it. Nevertheless, this kind of solution to metadata sharing may be technically difficult to accomplish properly. Regardless of the way this functionality is implemented, care should be taken that all the service providers are assured that metadata will not be exploited or robbed.

Many answers seem to support quite strongly the autonomy of individual subject gateways. For example, no centralised national databases or brokers are viewed as necessary for participating countries. Furthermore, it was seen that Renardus should broker directly to individual subject gateways and already existing brokers (like RDN [4]). Thirdly, the possibilities to build own user interfaces to the system and to configure the local user interface so that the service provider can decide which gateways are accessed received very strong support. The averages for these questions are the highest ones in the architectural part of the questionnaire (for A8 4,27 and for A9 3,9).

The level of normalisation of data is of course dependent on the architecture of the pilot system. Some kind of normalisation is needed in all the candidate architectural solutions [5]. The normalisation procedure should convert the syntax and format used by the information gateways forming Renardus to at least the chosen search result presentation format. Also particular search protocols (like LDAP) require the data to be presented to the search client in a particular way. In addition, when the broker system is used for metadata sharing it should have specified import and export format and syntax. In the case of Renardus, almost unanimous opinion of the respondents is that the system should have Dublin Core (DC) [6] as semantics and XML-based Resource Description Framework (RDF) [7] as syntax of metadata records.

Despite the architectural solution of the system, some search queries will certainly yield duplicate records to the search result lists provided by the broker. Although the respondents are aware of this problem they do not consider it as very severe at this stage of the project. The pilot system should simply list all the metadata records accompanied with information about the gateway they were retrieved from. Perhaps the eventual system should try to merge the duplicates to as comprehensive resource description as possible. In this ideal situation, the user would also be directed to the closest copy of the resource and she could choose whether she wants to see all the duplicates or not.

1.2 Semantic interoperability and multilinguality

In European information gateway project with many participants from different countries and organisations, semantic interoperability and multilinguality become important but difficult goals to achieve. Perhaps the easiest task in this context would be to give the end-user an opportunity to choose the language of the user interface. This option was strongly supported by the respondents (question B4 with an average of 4,36). However, multilinguality is not a top priority in the pilot system. According to the technical annex, the operational pilot service developed during the project will, at first, only provide an English user-interface to Renardus. Furthermore, it will not support multi-lingual indexing and searching. [1, p. 5.] Perhaps the service providers could do the translation of Renardus user interface after the pilot project. This is possible if they are given the opportunity to tailor the user interface of Renardus suitable for the special needs of their own user communities. However, as the results of the questionnaire suggest that language issues are important to the service providers, Renardus should be able to incorporate multilingual functionality in the future.

Mappings between different classification schemes are vital when cross browsing is implemented. The respondents favour this possibility over the possibility of one common classification scheme. Common scheme and mappings between different thesauri also received moderately strong support. After the pilot a common, maybe even multilingual, classification scheme should be considered. Conversions between different classification schemes can be done, for example, by using RDF as a mapping syntax.

The need for mappings between metadata formats in the context of Renardus is very obvious. Regardless of the architectural solution, some kind of normalisation of metadata is always needed [5], if only to provide a consistent search interface and view of the search results to the end-user.

The possibility to use automatic translation technologies in the Renardus system was not valued very highly by the respondents. Perhaps because of the still immature state of these technologies, it was seen that there is no need to automatically translate resources or free text fields of metadata records. Nevertheless, there are already services on the Web that handle translations of documents from one language to another. For example, Babel Fish [8] can make crude translations to both directions between English, French, German, Italian, Portuguese and Spanish. It is likely that the respondents considered making translations to be the responsibility of the resource creator.

1.3 Searching and browsing

The respondents feel that at least fairly developed search functionalities need to be incorporated into the pilot system. Cross browsing should also be implemented at some level as this functionality got strong support in the question C2 (with the average of 4,09). However, three of the respondents pointed out in their answers to question C4 that trying to implement very sophisticated cross browsing might be too difficult at this stage of the project:

> *Searching is probably easier to achieve in the short term, cross browsing of gateways is potentially complicated business. (ILRT)*

> *In theory both should be equally important but sophisticated cross browsing will be difficult to realise because extensive mappings between all the different classification schemes will be needed. (KB)*

The ability to restrict the search by choosing the language of the resource received very strong support (in question C1.1). The service providers valued also the possibilities 1) to restrict the search with the properties of metadata (like metadata creating date, creator organisation of metadata, etc.), 2) to choose which gateways are accessed with a search query and 3) to use the resource location (i.e., country) as a search criteria. All these options received fairly strong support and should most preferably be implemented in the pilot system. As revealed by the answers to the open question C.3, the possibility to restrict the search by using metadata fields as search criteria was generally appreciated by the respondents. It was perhaps supposed that the ability to do free text searches from all the record fields would be incorporated into the broker system as a default feature.

Only five of the respondents did give an exact answer to the question of acceptable delay for obtaining the search results (C.5.). These answers varied from 2 seconds to 15-20 seconds. The answer given by DTV seems illustrative in this context:

> *Depends on the actual situation and what results the user have asked for. In general for a simple search without any advanced features – a few seconds. (DTV)*

In addition, NETLAB presented an interesting solution to the problem of too long waiting periods:

> *As soon as there is a result of some kind, it will show up, no matter if the search is complete or not. The users will be able to choose how long they are prepared to wait for answers from all the gateways. (NETLAB)*

This solution is rather attractive in a sense that it gives the end-user the possibility to explore the already found resources while the system operates in another browser window in the background trying to find still more relevant records. Nevertheless, this scenario would not allow effective sorting of the search results by using relevance ranking algorithms.

1.4 Management

Monitoring end-user behaviour by collecting statistics was regarded as a central feature of the broker service (question D.5. with the average of 4,18). The service providers and possible Renardus administrators will be able to develop a better service by analysing the usage of the broker system. Usage statistics also provide information gateways with valuable information concerning the information needs and actual search behaviour of end-users. Therefore, more specific questions must be asked about what kind of data the system should include into its log files.

Another feature that the service providers seem to value is automatic notification of users of the new resources added to Renardus. This means that some kind of crude user profiling may be implemented in the pilot system. For example, the web bookstore Amazon [9] has a feature that users can choose to be notified by email when new records of books dealing with a theme that the user is interested in are added to the collection database. Another way to implement this functionality is to provide current awareness service in the context of Renardus. This service could be given by organising new links available in different information gateways in a common classification scheme that users can browse.

More developed versions of user or group profiles would require user authentication. However, the service providers seem to think that neither developed user profiles nor user authentication are very necessary at the pilot phase. In addition, there was no need for access control to some resources or resource descriptions. Nevertheless, when thinking about the eventual system, possibilities to build a personalised user interface and adding own bookmarks should be considered. If authentication of the users is needed in the future, the most popular way to do this is by user names and passwords. IP numbers and digital certificates are seen as less favourable methods, perhaps because they might be more restrictive to use and more difficult to implement. One should also keep in mind that authentication ought to be an optional feature of the eventual system. Authentication would only be required from those end-users that prefer to have a personalised user interface.

Automatic processing of resources turned out to be quite unpopular. Firstly, full-text indexing did not receive much support. Secondly, automatic tools for resource checking or classification are not seen as desirable. Thirdly, the ideas of converting resources on the fly to other file formats or to a packed form were even less popular. The only kind of automatic tool that gained some popularity was validity checking of hypertext links. That functionality should be considered after the pilot.

Different kind of computer aided metadata tools might ease the task of service providers. These tools could search automatically for candidate resources, extract important keywords from documents, propose classification codes, etc. As the results suggest, it is not seen as necessary that the Renardus broker provides these tools. If a purely distributed architecture is implemented, cataloguing new resources will take place at the information gateway ends of the broker system. It is likely that these kinds of tools are already used by some information gateways.

The most popular update frequency of metadata records to a central database or some other kind of metadata repository or index is daily or, more precisely, nightly (five answers). There was also support for updating metadata on a weekly basis (two answers). The real time solution would be to update the repository or centralised index every time a new record has been added to a gateway (the solution presented by NETLAB) or when the metadata of existing records changes (as suggested by DTV). However, most of the respondents probably thought when answering the question that this solution could cause too much network traffic or make the information gateways too busy in updating themselves in a time of day when the end-users mostly use the service. Nevertheless, Renardus broker will probably be, i.e., because of the collaboration with IMESH toolkit project [10], in intercontinental use in the future. Thus, there will be no right time of day to make the updates. From the Renardus perspective, however, the needs of European users should be prioritised.

In their present form, information gateways mostly contain descriptions of freely available digital on-line resources. The respondents did not support the idea of brokering to national bibliographies, other reference databases or to the collections of archives and museums in the Renardus context. However, collaboration with archives and museums is mentioned in the technical annex [1, p. 9]. Perhaps this kind of collaboration could be started after the actual project period is over. Commercial resources available, for example, in the databases of scientific publishers may also be incorporated into the eventual broker system in the future.

According to the respondents, Renardus should support at least Z39.50 and WHOIS++ -protocols (both were mentioned in 8 answers). Other protocols that were taken up in the answers are LDAP (3 answers), HTTP (2

answers) and CIP (1 answer). It was perhaps thought by most of the respondents that the support for HTTP in Renardus is so self-evident that it does not need to be mentioned at all.

One question that is only partially technical by nature is the relationship between existing gateways and the Renardus system. The service providers are seen to be responsible for interoperability with Renardus both during the project period and after it. One remarkable feature of the question concerning interoperability with Renardus on the longer term (question 9.2.) is that four of the respondents did not answer it at all. It was perhaps thought that this is an issue that should be thoroughly discussed when developing organisational and business models for Renardus. However, as can be seen from answers to the open question D.13, many respondents feel the need to have financially supported maintainer or co-ordinator organisation so that the continuity of the system after the project can be ensured.

In this respect, answers to the open question concerning the means to make sure that interoperability with individual gateways and Renardus continues after the project period (question D.13.) were also interesting. On the one hand, some respondents thought that it is the Renardus end that must change when a need arises.

> *The Renardus architecture must be modular so that different protocols and interfaces to gateways can be added later. (CSC)*

> *It must be flexible enough not to hinder technical changes but to adapt to it. (DDB)*

> *The central system should be capable of adjusting to changes made by participating gateways. New information gateways should be encouraged to work within Renardus framework - where they don't, the central system should adjust. (UKOLN)*

On the other hand, there were also respondents that supported more the option that the individual information gateways should change or comply with the demands of Renardus system:

> *[The interoperability is ensured b]y using standardised Z39.50 profile, which is infrequently changed. (DTV)*

> *Not! It is the service providers that should guarantee their interoperability with Renardus, if they cannot (or no longer) meet Renardus basic requirements they cannot join or continue participation. This does not mean that the Renardus system itself should not be upgraded and developed when necessary… (KB)*

There seems to be a need to find some kind of balance between the two options mentioned above. When planning the architecture it should be kept in mind that it is preferable for Renardus to be as open and modular as possible to support interoperability and incorporation of new kind of technological innovations. However, it is also clear that individual information gateways should comply with the basic demands set by the Renardus system.

1.5 Transparency of the system

From the service providers' point of view, the Renardus system should be quite transparent to the end-user. First of all, the search result lists should disclose the gateway that originally provided the record for Renardus. This can be done, for example, by showing the miniature logo of the gateway or gateways that contain a certain record. These logos can at the same time be links pointing to individual gateways. The end-user should also have a possibility to view the history of a specific search. For example, she should be able to find out which information gateways or national gateways are brokered to and how many relevant resources are found from each gateway. Thus, it becomes possible for the end-user to notice that a certain individual gateway fulfils her

needs best and she can turn to use just this individual gateway that most probably is faster than the Renardus system as a whole.

End-users are sometimes (perhaps even very often) more interested in satisfying their information needs than to know which information gateways have provided the metadata they are viewing. However, service providers themselves did not hold the possibility of choosing not to be aware which gateways have provided the record as a very important feature of the broker system (question E.2. with the average of 2,91, which is the lowest one in part E). Perhaps some service providers fear that if this option were possible, individual subject gateways would loose their identity and become only faceless metadata repositories for Renardus.

In the light of the answers given by the service providers, it might even be discussed whether Renardus should have a user interface of its own at all. In the pilot phase, a common user interface to Renardus will be necessary for development and testing purposes but in the long run it may not be needed.

1.6 Other Comments

As revealed by the comment section of the questionnaire, the respondents have sometimes experienced problems in interpreting some of the questions. This is a methodological problem affecting the validity and reliability of the results. However, there seem to be no questions that more than one respondent complained to be incomprehensible.

Due to the small amount of respondents, this survey cannot be seen as a qualified statistical analysis. However, since the answers reflect the collective opinion of the organisations that are involved in Renardus, they have each more weight than the answers of individual persons in the service provider organisations would have. This means that our unit of analysis has been organisation, not the individuals that act in them.

Because of the tight time schedule of the questionnaire some organisations also experienced difficulties to meet the deadline. For example, UKOLN gave the following comment:

> *Given the federated nature of the RDN there will need to be time for liaison with individual gateways on how their data is accessed by Renardus, we cannot achieve that in the time available for this questionnaire. We do know that gateways will not want to commit significant additional effort to participating in Renardus once the initial configura tion has taken place.*

However, there is not a lot of time available to make the basic functional and architectural solutions for the Renardus system. Thus, we as a project group have to make decisions, even if some of them turn out to be wrong ones in the long run.

2 USE CASE SCENARIOS

Renardus needs to capture user requirements within a short space of time, both the requirements of service providers who are participating in Renardus and end-users. Main service provider requirements are to be found in the present document. These requirements can be articulated further in meetings and via feedback on mailing lists. However, it is more difficult to involve end-users given the restraints on time and effort.

Nevertheless it is most important to ensure the design and architecture of Renardus supports the goals of the end-user. It is not feasible to undertake a major requirements gathering exercise, the project is not able to present prototypes to end-users to elicit detailed requirements, similarly we do not feel we are in a position to undertake face to face meetings with end-users in focus groups or such like.

In these circumstances one possible approach to assist the process of identification of requirements is to construct use cases or scenarios. If this is felt to be worthwhile then a number of use cases might be elaborated over the next few weeks to aid decisions on functionality and architecture. The insight gained by this approach would supplement the conclusions drawn from the questionnaire surveys, the review of existing broker models and the scoping document. All of these will provide valuable input into the architectural design.

2.1 What is a use case?

The term 'use case' was popularised by Jacobson [11] as a label for a more or less formal description of how an actor uses a system. Since the introduction of the concept there have been many definitions of use cases, these are characterised by Fowler [12]:

> *In essence, a use case is a typical interaction between a user and a computer system. Take the word processor I'm using to write this book. Two typical use cases would be "make some text bold" and "create an index. From just those examples, you can get a sense for a number of properties of use cases: a use case captures some user-visible function, a use case may be small or large, a use case achieves a discrete goal for the user.*

Within software engineering practice, particularly object oriented software development, use cases play a significant role in the design process and their use has been formalised to a greater or lesser extent depending on the principles of software design being followed. For example, the formalism of use cases can include diagrammatic representation in Unified Modelling Language (UML) [13], the use of templates to describe use cases, and the building up of relationships and links between use cases.

It is worth briefly distinguishing use cases from requirements and attribute/methods models [14]. Whereas a use case describes step-by-step the interaction of actor and system, the requirements specification would give a 'wish list' from the user viewpoint. For example, a requirements specification typically consists of statements such as 'The system should provide...', 'The system will need to provide…'. The attributes/methods model assumes a specific design even at the requirements gathering stage describing how different components of the system will behave.

The term 'scenario' is often used in relation to use cases. The 'scenario' is the detailed interaction, which is described within the use case. The use case supplements the scenario with additional information recording triggers, preconditions, related information, variations and extensions. Given the constraints on time and effort within Renardus we will be concentrating on developing scenarios rather than attempting to follow the more formal recording mechanisms of use cases.

2.2 Renardus methodology

What we intend to do here in Renardus will of necessity be informal, looking at use cases to help us identify key areas for attention. We will devise a simple template and produce a short textual description for each use case.

This will include the 'normal' user system interaction but will also need to capture 'exceptions'. It is often the exceptions that reveal the most interesting aspects of an interaction.

Our objectives will be to

? ? Identify functional requirements

? ? Aid communication between the project partners

? ? Stimulate thinking about the effect of requirements on architectural choices

It would be useful to provide high-level use cases for each of the significant user interactions with Renardus. As an initial step we have identified a number of possibilities emerging from the first draft of replies to the service provider questionnaires. For each of these interactions a use case might be described. By developing these use cases, and presenting them to the system developers in WP2, the architectural impacts will be revealed. Then if any particular use case has significant architectural implications it will need to be developed more fully.

Use cases can be divided into those where the actor is

? ? End-user

? ? Service provider (local gateway)

? ? The system (Renardus)

2.2.1 End-user as actor

Some possible scenarios that might be developed where the actor is the end-user have been suggested by replies to the service provider questionnaire. These suggest the end-user will be interested in

? ? Choosing the language of the user interface

? ? Searching by language of the resources

? ? Choosing which remote services can be accessed from the local gateway, i.e., user defines her own 'landscape' or search space

? ? Having searches automatically routed to gateways that can satisfy the search

? ? Retrieving resources by means of a 'subject approach' by entering subject search terms

? ? Retrieving resources by means of a 'subject approach' by browsing subject terms

? ? Getting automatic notification of new resources (current awareness)

? ? Storing a user profile (this might be used in definition of user's information space and in current awareness)

? ? Selecting a ready-made 'group' user profile

2.2.2 Renardus as actor

Scenarios where the Renardus system is the actor include

? ? Collecting usage statistics

? ? Ensuring local gateways are not overloaded by queries they cannot satisfy

? ? Storing user and group profiles

? ? Brokering to commercial resources

2.3 Use case template

The template proposed for use in Renardus is adapted from the template made available by Alistair Cockburn on his web site [15]. It is a simplified version of his template, with some changes in terminology to make it more applicable to Renardus.

Use Case name <active verb phrase>

CHARACTERISTICS

Goal <longer statement of goal>

Preconditions <what is already state of the world, if appropriate>

Success end condition <state of the world on successful completion>

Failed end condition <state of world if goal abandoned>

Primary actor <role name: end-user, Renardus system, local gateway, local broker)>

MAIN SUCCESS SCENARIO

Step 1

Step 2 <list steps of interaction from initiation to successful completion>

EXTENSIONS <list the variations that might occur in exceptional conditions, each referring to steps in the main scenario >

Extended Step 1.1

Sub.use case name

Condition <why extended?>

Extended scenario < what are steps of extended scenario?>

Extended step 1.2

Extended step 2.1

Extended step 2.2

RELATED INFORMATION

Priority <how critical to Renardus>

Performance Target <how long this interaction should take>

Frequency <how often it will occur>

OPEN ISSUES

<notes and comments>

2.4 Example use cases

Given that it is not feasible to contact end users direct, scenarios will be constructed by informed parties. It is useful to draw on existing functional specifications, such as the Agora HLMS Functional Specification [16], as well as project documents in order to identify key interactions.

2.4.1 End-user defines landscape for searching or browsing

Use Case name

End-user defines landscape for searching or browsing

CHARACTERISTICS

Goal

The end-user wishes to select relevant local gateways as a target set for searching or browsing

Preconditions

There will be a number of local gateways accessible from Renardus. Each gateway will be described by a service profile. The service profiles will be accessible for searching and browsing. The system will know which services are 'on-line' and which are 'out of service'.

Success end condition

End-user can enter search terms to restricted number of gateways or can select services by browsing.

End user selects gateways to search by combining various criteria such as

? ? Language

? ? Subject

? ? Availability (service is 'on-line', performance level acceptable)

? ? Region

? ? Known targets specified by user

Failed end condition

End user cannot effectively pre-select gateways

Primary actor = end-user

MAIN SUCCESS SCENARIO

Step 1

User selects 'define landscape' function

Step 2

The user can select gateways by means of browsing through various ‘pull down' menus. This would be appropriate for specifying a selection of

? ? Region

? ? Language

? ? Local gateway names

? ? Subject hierarchies

Step 3

The user additionally can initiate a search of service profiles (collection level descriptions) to select gateways that match

? ? By region

? ? By language

? ? By subject term

? ? By name

Step 4

The user is presented with a list of the gateways that match their selection criteria

Step 5

The user can save their selection of gateways for re-use

Step 6

The user then moves on to locate relevant resources by searching the selected gateways or browsing subject term lists for selected gateways.

EXTENSIONS

Extended Step 4.1

The user is informed if any gateways in their selection are 'out of service'

Extended Step 6.1

The user can re-define landscape at any time during a session

RELATED INFORMATION

Priority level

Desirable

Performance Target

<how long this interaction should take>

Frequency

User will frequently construct a new landscape depending on the nature of their search. Some users will tend to use saved landscapes.

OPEN ISSUES

? ? Will the user need to build a landscape for every search or browse of Renardus?

? ? Will there be a default landscape?

? ? Can a 'group landscape' be configured by a library or other intermediary?

2.4.2 End-user can choose a default landscape

Use Case name

End-user can choose a default landscape

CHARACTERISTICS

Goal

The end-user wishes to select all available gateways as a target set for searching or browsing

Preconditions

The system selects all participating services. The system will know which services are 'on-line' and which are 'out of service'.

Success end condition

End-user can search or browse all gateways by default.

Failed end condition

End user cannot select all gateways by default.

Primary actor = end-user

MAIN SUCCESS SCENARIO

Step 1

End user selects default to access all gateways

Step 2

The user is presented with a list of the gateways that are available.

Step 3

The user then moves on to locate relevant resources by searching the selected gateways or browsing subject term lists for selected gateways.

EXTENSIONS

Extended Step 2.1

The user is informed if any gateways in their selection are 'out of service'

Extended Step 3.1

The user can re-define landscape at any time during a session

RELATED INFORMATION

Priority = essential

Performance Target

<how long this interaction should take>

Frequency

User will frequently wish to use the default of searching across all services.

OPEN ISSUES

None

2.4.3 End user can maintain picture of their landscape or 'search space'

Use Case name

End user can maintain picture of their landscape or 'search space'

CHARACTERISTICS

Goal

The user will have information displayed on the screen, which informs them of which services are being searched. If they input a new search the system will display clearly which services they are searching.

Preconditions

Once a user landscape has been defined it will be characterised and displayed to the user to aid navigation.

Success end condition

The user will be able to navigate effectively

Failed end condition

The user will not be aware which services are being searched

Primary actor = end-user

MAIN SUCCESS SCENARIO

Step 1

User defines landscape

Step 2

Definition of landscape is summarised for user to confirm or amend

Step 3

Throughout following searching and browsing interactions the user's choice of landscape can be easily displayed

EXTENSIONS

Extended Step 3.1

If one of the services included in a selected landscape becomes unavailable mid-search then the user will be informed.

RELATED INFORMATION

Priority = desirable

Performance Target

<how long this interaction should take>

Frequency

During all interactions the user will be made aware which services are included in their 'landscape'.

OPEN ISSUES

How will a selected landscape be characterised? Will it, for example, be characterised by a name or by a list of all included local services?

3 CONCLUSION

The two most general conclusions of this deliverable are that, firstly, from the service providers' perspective Renardus should be an addition to their existing information gateway requiring only reasonably small adjustments to the current way of doing things. Secondly, from the end-users' point of view Renardus is ideally an intuitive, fast and accurate way of locating the information they need. These requirements may in some cases be contradictory. It should, however, also be remembered that end-users are a very heterogeneous group. Furthermore, as can be seen from the critiques of conventional user-studies, it is often not easy at all to find out what their real needs are (cf. [17-19]).

In the following we present more detailed conclusions of the service provider survey. The requirements are divided into five categories. The first three categories are based on the definition of the task plan [2, p. 3]:

> *[F]unctionality is divided into 'essential', 'preferable', and 'desirable'. Essential functionality must be implemented. Most preferable functionality should be implemented but there may be some trade-offs. Desirable functionality will only be implemented if it proves straightforward, or if there is sufficient effort left after essential and preferable functionality has been implemented.*

In addition there emerged two further categories: 'functionality **not** requested' and 'problematic areas requiring discussion'.

The chapter 2 of the present document shows that it is possible to carry the requirement analysis further by constructing detailed use case scenarios from the key requirement areas. Because this work will continue after the formal publication of this deliverable, it is not necessary but to briefly summarise it here. Use cases in the Renardus can be divided into those where the actor is a) end-user, b) service provider (local gateway) and c) the system (Renardus). Use case template for the purposes of the project is presented. The exemplary use cases given in the chapter are: 1) end-user defines landscape for searching or browsing, 2) end-user can choose a default landscape and 3) end user can maintain picture of their landscape or 'search space'.

The end-user survey results are collected to chapter 3.6. The end-user requirements cannot at this phase be categorised in a similar way as we have done with the results of the service provider survey.

3.1 Essential requirements

? ? Renardus must be based on a distributed architecture.

? ? Renardus must have centralised index of collection level metadata for query routing purposes. Another option would be to implement a centralised index with decentralised databases.

? ? Renardus must support Dublin Core semantics and XML-based RDF syntax for metadata records.

? ? Renardus must support at least Z39.50 and WHOIS++ protocols.

? ? Centralised index information must be updated frequently, preferably every 24 hours.

? ? Mappings between different metadata formats and classification schemes must be built to support searching and cross-browsing.

? ? Search functionality must be fast and searches can be restricted by all metadata fields, which should include

- ? ? Properties of metadata (creation date, creator name, etc.)
- ? ? Origin of the metadata (i.e., gateway or broker)
- ? ? Language of the resource

? ? Location of the resource (i.e., country)

? ? Renardus must implement cross browsing at some level.

? ? Service providers want Renardus to be transparent to the end-user, i.e., the end-user must always be made aware of the origin of the metadata records.

? ? It must be possible to make local changes, additions and customisations to the Renardus user interface (i.e., by translating it to other languages).

? ? Renardus must monitor end-user behaviour and collect usage statistics.

3.2 Preferable requirements

? ? Renardus should merge duplicate metadata records intelligently.

? ? Renardus should support sophisticated cross browsing.

? ? Renardus should check the validity of links to resources.

3.3 Desirable requirements

? ? Renardus should support LDAP and CIP protocols if needed.

? ? User interface should support different languages.

? ? One common classification scheme.

? ? Mappings between different thesauri.

3.4 Functionality not requested

? ? One common metadata repository for sharing metadata.

? ? Automatic translation of free text fields in metadata records or whole resource documents.

? ? Tool for the end-users to convert resources on the fly to another format or to a packed form.

? ? Automatic full-text indexing.

? ? User authentication.

? ? Access control to resources.

? ? Brokering to national bibliographies, other reference databases or to digital collections of archives and museums.

? ? Automatic tools for classification of the resources.

? ? Automatic resource checking tools that warn if metadata is out of date.

? ? Renardus to require national databases or brokers.

? ? Renardus to be European level master database.

3.5 Problematic areas requiring discussion

? ? What do centralised and distributed architectures mean more exactly?
How do the different technical solutions affect maintainability, search times, etc.?

? ? What kind of statistics should be collected about the usage of the broker system?

? ? How is the responsibility of maintaining Renardus shared after the project is finished?

? ? Which end should adapt more to changes, Renardus or the information gateway searched by the broker?

? ? Multilinguality and localisation were important concerns for the service providers, but TA is explicit about leaving multilinguality out in the pilot system development phase.

? ? How will the user interface to the broker be implemented? Will Renardus have a user interface of its own after the pilot?

? ? Metadata sharing is an important overall objective of the Renardus project, but many of the project partners are wary about it.

? ? Service providers oppose user authentication, but want to offer some personalised services.

3.6 End-user requirements for the broker system

The requirements from the end-user perspective are mostly analysed in deliverable 6.6. From the first formal draft of this deliverable [20] following end-user requirements can be found:

? ? The broker system should support also end-users that are not attached to fast LANs, but use ISDN or only modem.

? ? The user interface of the system should be as self-explaining as possible. Separate help screens are perhaps not needed at all.

? ? Searching and browsing have different functions to the end-users: searching is for finding specific information to a quite defined or articulated information need. The information needs directing browsing are more general in nature. The browse option is mostly used when a wider range of information is searched for.

? ? End-users prefer searching to browsing. They seem to want an intuitive, yet accurate and fast way for locating the information they need.

? ? Simple search is used more often than advanced search. Often the search is done with one search term only. Boolean operators or other advanced search functionalities are quite rarely used.

? ? The favourite metadata fields to restrict the search are keyword, author, title and description.

? ? The default presentation format of the search results should show 10 to 20 records per page.

? ? The primary target audience of the broker system is higher research and education community as a whole (including researchers, teachers and students).

? ? The end-users seem to appreciate the quality control and classifying work performed by the service providers. This feature of the service should be emphasised in the possible Renardus user interface.

PART IV - REMAINDER

REFERENCES

1. TA (1999), Annex 1 – "Description of Work". Date of preparation of Annex 1: 29.11.1999.

2. Gardner, T. (2000), Reynard: WP1 (Functional Model) Task Plan. Draft date: 12.1.2000.

3. Dovey, M.J. (2000), Renardus: Considerations towards determining an architectural model. Document date: 23.2.2000. (http://homes.ukoln.ac.uk/~lisrmh/renardus/archoptions-v1.htm)

4. RDN – Resource Discovery Network (http://www.rdnet.ac.uk/).

5. Sandfaer, M.(2000), Renardus Architectural Options. [Power point document presented at Renardus technical meeting in Lyngby, Denmark 29th of February 2000.]

6. Dublin Core Metadata Initiative (http://purl.org/DC/).

7. Resource Description Framework (RDF) (http://www.w3.org/RDF/)

8. Babel Fish (http://babelfish.altavista.com/).

9. Amazon.com (http://www.amazon.com).

10. The IMesh Toolkit: An architecture and toolkit for distributed subject gateways (http://ukoln.ac.uk/metadata/imesh-toolkit/).

11. Jacobson, I., Christerson, M., Jonsson, P., & Övergaard G.(1992), Object-Oriented Software Engineering: A Use Case Driven Approach, Addison-Wesley

12. Fowler, M. and Kendall, S.(1997), UML Distilled: Applying the Standard Object Modelling Language. Addison-Wesley. [Chapter 3 of the book can be found at http://www.awl.com/cseng/titles/0-201-32563-2/umldist-chap3.html]

13. Richter, C.(1999), Designing Flexible Object-Oriented Systems with UML. Macmillan.

14. Use Cases vs. Requirements vs. Attributes/Methods. August, 1997 (http://ootips.org/use-cases-vs-requirements.html).

15. Cockburn, A. (1998), Basic use case template, 1998. (http://members.aol.com/acockburn/papers/uctempla.htm)

16. Newton-Ingham, G., Palmer, D. Kay, D & Smith, M. (1999), Agora Hybrid Library Management System (HLMS) Release 1 Specification, public draft (http://hosted.ukoln.ac.uk/agora/documents/agora_spec_public1.doc)

17. Dervin, B. (1992), From the Mind's Eye of the User: The Sense-Making Qualitative-Quantitative Methodology. In Glazier, J.D. & Powell, R.P.(eds.), *Qualitative Research in Information Management*, pp. 61-84. Englewood, CO: Libraries Unlimited.

18. Wilson, T.(1994), Information Needs and Uses: Fifty Years of Progress? In Vickery, V.C.(ed.), *Fifty Years of Information Progress*, pp. 15-51. London: Aslib.

19. Tuominen, K., User-Centered Discourse: An Analysis of the Subject Positions of the User and the Librarian. Library Quarterly 67 (1997): 4, pp. 350-371.

20. Becker, H.J., Klaproth, F. & Lepschy, P.(2000), Survey of End-User Surveys. First formal version of Renardus Deliverable 6.6, dated 2000-03-27.

ANNEXES

ANNEX A: Questionnaire for the service providers

The purpose of this questionnaire is to collect service providers' views on what participant services want as regards future functionality in the Renardus system. The functionalities and traits that the respondents see as essential are strong candidates of implementation in the pilot system. However, this questionnaire also aims to map out what kind of functionalities and/or value-adding features should be incorporated into the eventual system.

The questionnaire comprises mostly of statements. You should choose one of the options from 5 to 1 (5=essential, 4=preferable, 3=desirable, 2=not necessary, 1=should never be implemented or 5=strongly in favour, 1=strongly against) for each question. Each participant of the Renardus project fills in only one form so that there should be a consensus in your organisation concerning the content of the answers. There are a few open questions after each theme and a space for comments at the end of the questionnaire.

Results of the questionnaire will be analysed in deliverable 1.2. (user requirements for the broker system) that is due to be ready in March. The completed questionnaires should be sent back by email to Kimmo Tuominen (ktuomine@csc.fi) no later than 21st of February 2000.

A. Distributed/centralised system	***5***	***4***	***3***	***2***	***1***
1. Renardus should be one European master database to which all new records from participating information gateways are routinely copied	☐	☐	☐	☐	☐
2. Each European country has their own information gateway database (not broker) containing only national resources to whom the system primarily brokers	☐	☐	☐	☐	☐
3. Distributed searching/browsing system with no central databases	☐	☐	☐	☐	☐
4. Each European country has their own information gateway broker containing only national resources to whom the system primarily brokers	☐	☐	☐	☐	☐
5. Already existing broker systems are brokered to by the Renardus system	☐	☐	☐	☐	☐
6. Search query routing based in centralised indexes is used so that all search questions are not broadcasted to all information gateways	☐	☐	☐	☐	☐
7. In case of multiple URIs of the same resource, the system directs users towards the closest (in network terms) copy of the resource	☐	☐	☐	☐	☐
8. Possibility to build own user interfaces to the system tailored to the service needs of certain subject area/country	☐	☐	☐	☐	☐
9. Possibility to configure what remote information gateways are accessed from the local user interface	☐	☐	☐	☐	☐
10. Common metadata repository from which one can copy records to one's own information gateway	☐	☐	☐	☐	☐

11. What data import/export format (i.e., MARC, DC or GILS) and syntax (i.e., RDF, plain XML) should be used between the possible Renardus central metadata repository and/or existing information gateways/brokers?

12. How should duplicate records be handled by the system?

B. Multilinguality/semantic interoperability	***5***	***4***	***3***	***2***	***1***
1. One common multilingual classification scheme	☐	☐	☐	☐	☐
2. Mappings between different classification schemes	☐	☐	☐	☐	☐
3. Mappings between different thesauri	☐	☐	☐	☐	☐
4. Possibility to choose the language of user interface	☐	☐	☐	☐	☐
5. Automatic translation					
- of free text fields (i.e., abstracts) of the metadata record	☐	☐	☐	☐	☐
- of whole documents/resources	☐	☐	☐	☐	☐

C. Browsing/searching	***5***	***4***	***3***	***2***	***1***
1. Search can be restricted or expanded by					
- language of the resource	☐	☐	☐	☐	☐
- location (i.e., country) of the resource	☐	☐	☐	☐	☐
- information gateway	☐	☐	☐	☐	☐
- metadata properties (metadata creator organisation, date of metadata creation, etc.)	☐	☐	☐	☐	☐
2. Cross-browsing (browsing simultaneously different information gateways)	☐	☐	☐	☐	☐

3. What advanced search features should be available in the pilot system?

4. Which are more important in the pilot service, search or browsing functions? Why?

5. What is an acceptable delay in seconds for obtaining the search results?

D. Management of the system	***5***	***4***	***3***	***2***	***1***
1. Full-text indexing of the resources (not just metadata)	☐	☐	☐	☐	☐
2. User authentication[1]	☐	☐	☐	☐	☐
- user name/password	☐	☐	☐	☐	☐
- IP numbers	☐	☐	☐	☐	☐
- digital id (certificates)	☐	☐	☐	☐	☐
3. Access control for					
- certain resources	☐	☐	☐	☐	☐
- certain resource descriptions	☐	☐	☐	☐	☐
4. User/group profiles[2]	☐	☐	☐	☐	☐
- personalised user interface	☐	☐	☐	☐	☐
- notification of new resources based on user preference	☐	☐	☐	☐	☐
- personal bookmarks	☐	☐	☐	☐	☐

[1] If you choose 2 or 1, answers to the clarifying statements about authentication methods are not necessary

[2] If you choose 2 or 1, answers to the clarifying statements about the uses of profiles are not necessary

5. User behaviour is monitored and statistics are collected for the benefit of service providers. ☐ ☐ ☐ ☐ ☐

6. Documents can be converted on the fly
   - to another format (html -> pdf, etc.) ☐ ☐ ☐ ☐ ☐
   - to packed form (zip, tar, etc.) ☐ ☐ ☐ ☐ ☐

7. Renardus should also broker to
   - national bibliographies ☐ ☐ ☐ ☐ ☐
   - other reference databases ☐ ☐ ☐ ☐ ☐
   - collections of archives, museums, etc. ☐ ☐ ☐ ☐ ☐
   - commercial resources ☐ ☐ ☐ ☐ ☐

8. Tools for
   - automatic classification of resources ☐ ☐ ☐ ☐ ☐
   - validity checking for links to resources ☐ ☐ ☐ ☐ ☐
   - automatic resource checking to warn that metadata may be out of date ☐ ☐ ☐ ☐ ☐

9. Service provider is responsible for interoperability with Renardus
   - for the project period ☐ ☐ ☐ ☐ ☐
   - on the longer term ☐ ☐ ☐ ☐ ☐

10. What would be a reasonable frequency for updating metadata from individual information gateways to central master database or some other kind of metadata repository?

11. What standard protocols should Renardus (i.e., LDAP, WHOIS++, Z39.50) and existing gateways or brokers support? What APIs should be build?

12. How should the system be able to adjust to technical changes made in participating information gateways or to new information gateways joining Renardus?

13. What should be done so that interoperability with Renardus continues after the project period?

E. Transparency of the system to the user ***5 4 3 2 1***

1. The name and URI of the information gateway that originally provided the record is displayed in the search result lists ☐ ☐ ☐ ☐ ☐

2. User can choose to be not aware that he/she is browsing/searching multiple information gateways at the same time ☐ ☐ ☐ ☐ ☐

3. Possibility to have information of the way a specific search has proceeded (what countries / national gateways / information gateways are brokered to, how many relevant resources found from which gateway, etc.) ☐ ☐ ☐ ☐ ☐

F. Other comments

Other comments concerning the broker architecture and functional requirements of Renardus?

Thank you for your answers!

ANNEX B: Answers to closed questions

	ALUH	BnF	CSC	DDB	DTV	ILRT	JyU	KB	Netlab	SUB	UKOLN	AVG	STDEV
A1	1	1	5	1	2	3	1	5	3	2	2	**2,36**	**1,5**
A2	1	1	4	2	2	2	2	3	2	1	2	**2**	**0,89**
A3	5	4	1	4	5	4	5	2	4	5	3	**3,82**	**1,33**
A4	2	4	3	2	2	2	3	2	2	1	4	**2,45**	**0,93**
A5	4	4	3	2	2	4	4	4	2	5	4	**3,45**	**1,04**
A6	4	5	3	4	3	4	4	3	4	3	3	**3,64**	**0,67**
A7	3	2	3	3	2	5	3	3	2	3	3	**2,91**	**0,83**
A8	4	4	4	5	5	5	4	5	4	4	3	**4,27**	**0,65**
A9	4		4	5	4	5	4	4	2	4	3	**3,9**	**0,88**
A10	2		5	2		3	1	3	2	3	2	**2,56**	**1,13**
B1	2	3	4	4	4	2	2	4	4	4	2	**3,18**	**0,98**
B2	4	5	4	4	5	4	3	4	5	3	3	**4**	**0,77**
B3	3	5	3	5	3	2	2	4	2	2	3	**3,09**	**1,14**
B4	5	5	4	4	3	4	5	5	4	5	4	**4,36**	**0,67**
B5.1	2	2	3	3	2	3	2	3	2	2	3	**2,45**	**0,52**
B5.2	1	2	2	2	1	3	1	1	2	2	2	**1,73**	**0,65**
C1.1	4	5	5	4	5	4	4	5	4	5	5	**4,55**	**0,52**
C1.2	5	2	4	4	5	3	4	3	4	4	2	**3,64**	**1,03**
C1.3	4	5	4	2	3	3	4	5	3	5	4	**3,82**	**0,98**
C1.4	4	5	4	4	5	3	4	5	5	5	2	**4,18**	**0,98**
C2	4	5	4	4	4	4	4	3	5	4	4	**4,09**	**0,54**
D1	2	3	2		5	3	1	3	3	3	3	**2,8**	**1,03**
D2	1	1	3	4	2	4	1	3	2		3	**2,4**	**1,17**
D2.1		1	3	4		4		4			3	**3,17**	**1,17**
D2.2		1	3	4		3		2			2	**2,5**	**1,05**
D2.3		1	3	2		2		2			2	**2**	**0,63**
D3.1	2	1	2	2	2	3	2	1	2			**1,89**	**0,6**

D3.2	1	1	1	2	3	3	2	1	3			**1,89**	**0,93**
D4	4	3	3	4	3	4	4	3	3	4	3	**3,45**	**0,52**
D4.1	4	3	3	4	2	4	4	4	2	4	3	**3,36**	**0,81**
D4.2	4	3	4	4	4	4	4	4	4	4	3	**3,82**	**0,4**
D4.3	4	3	2	4	2	4	4	3	2	4	3	**3,18**	**0,87**
D5	4	4	5	4	3	4	4	5	3	5	5	**4,18**	**0,75**
D6.1	1		3		2	2	1	2	2	3		**2**	**0,76**
D6.2	1		2		2	2	1	2	2	3		**1,88**	**0,64**
D7.1	1		1	4	2	4	1	1	2	4		**2,22**	**1,39**
D7.2	1		1	4	2	4	1	1	2	5		**2,33**	**1,58**
D7.3	1		1	4	2	4	1	3	2	5		**2,56**	**1,51**
D7.4	3		3	4	2	4	2	3	2	4		**3**	**0,87**
D8.1	1	2	2		1	4	1	3	2	3	4	**2,3**	**1,16**
D8.2	5	4	5	4	1	1	5	5	2	5	2	**3,55**	**1,69**
D8.3	4	4	4	4	1	1	4	3	2	5	3	**3,18**	**1,33**
D9.1	1		5	4	5	4	1	5	5		3	**3,67**	**1,66**
D9.2	4		5	3		2	5	5			3	**3,86**	**1,21**
E1	2	5	3	4	5	3	2	4	5	5	4	**3,82**	**1,17**
E2	4	2	4	4	2	3	4	4	2	2	1	**2,91**	**1,14**
E3	4	4	5	4	5	4	4	1	5	5	3	**4**	**1,18**
	2,98	**3,23**	**3,22**	**3,61**	**3,03**	**3,36**	**2,9**	**3,22**	**2,95**	**3,84**	**3,03**	**3,16**	**0,97**

ANNEX C: Answers to open questions and comments

A.11

ALUH *ROADS (IAFA)-template, DC*

BNF DC/RDF/XML

CSC DC/RDF/XML, XSLT

DDB Qualified DC. RDF/XML should be decided within the project.

DTV DC/RDF

ILRT May have to be able to cater for a variety of formats in order to ensure widest participation.

JYU Of course all known formats used in subject gateway services should be considered in this context – however, the starting point could be primarily the formats used by gateways included in the pilot version of the broker-service (e.g. Dublin Core and ROADS templates and the syntaxes compatible with them).

KB DC/RDF/XML or plain XML

NETLAB DC/RDF. Renardus will develop a metadata solution.

SUB DC/RDF/XML

UKOLN DC/RDF/XML

A.12.

ALUH Duplicates are not a serious problem. It should be possible for user to select which record she wants.

BNF All the duplicate records have to be managed by the system but the system can display one depending on a criteria (i.e. date of the last update).

CSC All records should be listed. Unification is technically very difficult.

DDB Provided that there is no master database and no master metadata pool, the system must route user queries towards the single information/subject gateways according to the content of the user queries. That means the system must have a small pool of metadata, appropriate to assign the queries. In this view the handling of duplicate records is the business of the single gateways.

DTV It's a hard problem, make it user visible and try to merge duplicates.

ILRT Users should be given the choice of whether or not they want to see duplicate records. If they choose to see duplicates they should be linked together in some way.

JYU Duplicate records should not be a problem. It may be, however, useful to construct a system which, in case of duplicate records, shows the information seeker the record produced in his/her own country, if such a one exists.

KB Integrate the information in one record. Descriptions in more than one language of the same resource can be used to provide multilingual functionality.

NETLAB This is not top priority. We will not raise the issue until users complain over duplicates.

SUB All listed, with a visible link (i.e. logo) to the source gateway. Listing has to be discussed.

UKOLN Display duplicate records.

C.3.

ALUH Search by main fields: title, country, keywords, resource type.

BNF Language of the resource, language of the record, date of last update, keyword in subject.

CSC Boolean operators, truncation and possibility to restrict the query to specific fields.

DTV Boolean, fields, free text, refine.

JYU Central field searches: author, title, keywords.

KB Search on specified metadata fields (e.g. author, title, keyword etc.) should definitely be possible. Also the possibility to select a subject category from the classification scheme and search within that category (combination of browse with search) would be useful.

NETLAB Search support, index, the possibility to search all common fields separately as well as in a combination, free of choice.

SUB Language, country, subject.

C.4.

ALUH Search.

BNF Browsing by theme, it means we need to define an arborescence.

CSC Search is more feasible technically. Cross-browsing would be nice, but may be difficult to Implement in the near future.

DTV Both are necessary.

ILRT Searching is probably easier to achieve in the short term, cross browsing of gateways is potentially complicated business.

JYU Search functions.

KB In theory both should be equally important but sophisticated cross browsing will be difficult to realise because extensive mappings between all the different classification schemes will be needed.

NETLAB The question seems irrelevant, at least as we interpret it. If we hadn't intended to at least try to use both, we didn't need this project. If we only wanted to make a search option, we could have used ROADS. What benefit could we gain from the answers to this question? We couldn't see any, so that is why we didn't want it in the questionnaire in the first place.

SUB Search functions, because there can be analysed more details on the accuracy of the data.

UKOLN Equally important.

C.5.

CSC 4

DTV Depends on the actual situation and what results the user have asked for. In general for a simple search without any advanced features – a few seconds.

JYU 5

KB 2

NETLAB As soon as there is a result of some kind, it will show up, no matter if the search is complete or not. The users will be able to choose how long they are prepared to wait for answers from all the gateways.

SUB 15-20

UKOLN Unknown.

D.10.

ALUH Daily.

BNF Weekly.

CSC Preferably nightly, at least weekly.

DDB Provided that there is no central database and that the central metadata pool only serves to route queries, an update of metadata is only necessary if the key for the assignment has to be changed.

DTV As needed (i.e. when metadata changes).

ILRT Weekly.

JYU Daily.

KB Daily.

NETLAB As soon as a new record has been added.

SUB N/A

UKOLN Nightly.

D.11.

ALUH WHOIS++, Z39.50.

BNF WHOIS++

CSC In the order of importance: LDAP, WHOIS++.

DDB Z39.50, HTTP.

DTV Z39.50, CIP.

ILRT Z39.50, WHOIS++.

JYU Renardus should support all those three. They should be enough and they probably cover all the gateways initially included.

KB Minimally HTTP; API's depend on service provider's needs: make an available pool of APIs that gateways use.

NETLAB WHOIS++, Z39.50.

SUB LDAP, WHOIS++, Z39.50.

UKOLN Z39.50 (and possibly Whois++).

D.12.

CSC The Renardus architecture must be modular so that different protocols and interfaces to gateways can be added later.

DDB It must be flexible enough not to hinder technical changes but to adapt to it.

DTV By using standardized Z39.50 profile, which in infrequently changed.

ILRT As flexibly as possible, information gateways will be bound by their own funding constraints and development plans - extra effort to comply with Renardus should be kept as minimal as possible.

KB Not! It is the service providers that should guarantee their interoperability with Renardus, if they cannot (or no longer) meet Renardus basic requirements they cannot join or continue participation. This does not mean that the Renardus system itself should not be upgraded and developed when necessary. Conditions for this should be formulated in the Technical Implementation Plan.

NETLAB It will not adjust for the sake of a single gateway (that would be hard for the gateways already participating). However, it will follow the general technical development concerning gateways, since some of the participating gateways might be in the lead for this.

SUB Can not be answered now.

UKOLN The central system should be capable of adjusting to changes made by participating gateways. New information gateways should be encouraged to work within Renardus framework - where they don't, the central system should adjust.

D.13.

ALUH Co-ordinating group.

CSC A centralised European-level technical maintainer group (2-3 persons) should be created with funding from the participating organisations.

DTV You also need a budget and a maintaining organisation.

ILRT See previous question - the smaller the effort required and the greater the benefits to the gateways, the easier it will be to make sure gateways remain involved.

JYU A clear joint set of rules should help. After the project, a development group could be set up which sees to interoperability in the changing world.

KB Good organisation and commitment formalised in some sort of agreement for collaboration. Plans for future availability of the service and further development, as well as all funding and organisational aspects should be specified in detail in the Technical Implementation Plan.

NETLAB Find someone who can take charge of the project and finance it. Mobilise support among different interest groups as associations, gateways or national initiatives.

SUB Extensive co-operation, some kind of agreements?

UKOLN Use of relevant standards within Renardus. Promotion of the use of those standards outside of Renardus.

F.

BNF A.7. If we consider the point of view of the information viability, we can answer 2 (as we do in the questionnaire). If we consider the network point of view, we have to answer 5. For us, the viability of the information is the most important matter to consider.

A.10. Does it mean Reynard could offer the possibility to copy records from an information gateway to another? If yes, there are the problems of updating the duplicate records.

B.1. One common multilingual classification scheme would be a good feature but not essential. Which one to choose? Dewey seems not to be convenient.

B.2. We would like to mention the CENL project called MACS (Multilingual Access for Subjects) which manages the mapping between LCSH, RAMEAU and SWD.

C.3. We could also have Type of resources (i.e. discussion forums) but not for the prototype.

D.2. User authentication. Does it mean the users have always to enter a password? We don't agree with this option.

D.7. For us, this question concerns the purpose of the Reynard project and not the management system.

DDB From the view of DDB as an information gateway provider, there are three basics we should clarify first: Should there be a central database or metadata pool? For which kinds of search queries shall the system be prepared? Only subject based queries or queries for authors, titles etc. too? Which kinds of subject gateways are included? Do they provide the whole literature of this subject - or a special part? How are the subjects delimited? Do they provide only the current literature?

We don't think that a central database would be useful. DDB wouldn't be able to provide an update service. If you allow questions that are not subject oriented, it would be reasonable to route these primarily to the national information gateways. On the other hand, subject oriented queries should be primarily routed to that gateway which is specialised on the topic. You need to have a key for the assignment of the queries according to the contents of the single subject gateways. Those queries which cannot be routed to the subject gateways or which the subject gateways fail to answer are routed to the information gateways.

A.2./A.4. If you drop "to whom the system primarily brokers" the statement is preferable.

A.7. Restrictions of access and costs should be considered, too.

A.10. A standardised interface to copy records between the gateways would be preferable.

D.1. That's not the matter of the Renardus system but of the single gateways.

D.6. That's the matter of the single gateways.

D.7. You enumerate information gateways. Drop "also" and add subject gateways.

D.8. first point: Automatic classification is the matter of the single gateways.

JYU A.4. The material coming to Renardus from national gateway-systems does not have to be strictly limited to national material. It may also include high-standard international material found in the gateway.

D.2. User identification I not a very important feature in a public service which is accessible by anyone. Of course this may be necessary in user-specific customising of the service. The possibility of user identification must not be excluded technically either, in case the situation changes in future and Renardus becomes fee-based for users.

D.7. The search should not be extended to national bibliographies or other similar databases- Renardus focuses on Internet resources. It is, however, important that such information sources can at national level be described as references to Renardus. If need be, it should be possible to extend Renardus searching to high-standard commercial Internet subject gateways.

KB General: There should be a built in redundancy in the central service: when one server is down, mirrors and/or back ups should guarantee its continued availability.

Scalability: The Renardus architecture should be made available for national services which want to establish a national broker service 8the scenario specified in Question A4). Renardus can then broker to this national broker, which has the same underlying generic architecture.

A.1./A.3. The core of the Renardus service should be a central database. Distributed searching could be offered as a second (extra) option, to give users a chance to try other databases when the Renardus database fails to answer their question.

A.6. This depends on the answer to A1/A3. In a centralised system as we would like to see, with additional distributed searching options, it is not that essential, that is why we answered 3, but if distributed searching is an important part of the Renardus system this would be more essential for performance, so in that case it would be 4 or even 5.

B.1. This should be a very basic high level classification, f.i. the top level of the Nederlandse Basisclassificatie (Dutch Basic Classification).

D.2./D.3./D.4. We see this as an option for the longer term rather than as something to be realised in the initial service.

E.1. Not too prominently, taking up too much space and annoying the user. Small logos/icons?

E.2. Implement as user interface option.

NETLAB A.1./A.3. We wish to add what we see as two possible solutions for the Renardus project. One way is to have a totally distributed system, which could be useful as a starting point. The other way (that could come after the totally distributed system is to have a system that is distributed in one way and centralised in another. That is, certain metadata fields form the distributed services will be used in a central search system.

A.5. Question is unclear. Do you mean exclusively or inclusively? Your answer probably depends on how you interpret the question.

C.1. The last on about other metadata properties should be an open question, since there are a lot of interesting fields.

D.8. This have to be solved at gateway level and not at a Renardus level.

D.9. Responsibility in the longer term isn't going to be solved here. The answer for this depends on what will happen with the service in the future.

SUB A.7. Hint: Would be nice if this feature can be "switched on/off" by the user.

A.10. That's a point we have to discuss in detail, it' not done by this ranking! There are a lot of undiscussed an unsolved issues concerning the IPRs and the who and where should such repository stay. D.1. This is for _selected_ resources only.

D.2. Sorry we can't answer this because there are some relations the questions can't resolve to answer with a system 1-5. Let us discuss this in Copenhagen, please.

D.3. - " -

UKOLN Section A: RDNC would like to investigate the possible options for achieving benefits to the RDN from sharing records, centralised databases, query routing etc. in a European context. We see this project as an opportunity to investigate how collaboration with other gateways can provide an overall improved service to the end-user. For this reason we would support innovative solutions (i.e. more than cross searching using Z39.50 base line profile)

In order to this we do not wish to be prescriptive as regards architecture. Given the federated nature of the RDN there will need to be time for liaison with individual gateways on how their data is accessed by Renardus, we cannot achieve that in the time available for this questionnaire. We do know that gateways will not want to commit significant additional effort to participating in Renardus once the initial configuration has taken place.

Section B: The second part of question 5 (and a couple of other questions) imply, to me, that Renardus will deliver modified versions of the resources described by gateways. Is that correct? Section D. I've indicated that user authentication is desirable - in doing so I'm assuming that user authentication would somehow be an optional feature of the Renardus interface - in other words, I think it would be desirable to have authentication as an option! :-) I don't really understand question 3 here. At least, I don't understand how Renardus could implement access control for 'certain resources' - Renardus enables the discovery of resources - access control for resources is in the hands of the resource operators isn't it?? Again, question 6 seems to imply conversion of the resources that are described by gateways? Re: question 12, I think it is important to note that Renardus should be responsive to what gateways and services are already doing and should adapt in line with future changes - rather than expecting gateways to modify their services to fit in with a Renardus approach. Section E. Question 1 specifically mentions the use of a URI to acknowledge the source of a record. Acknowledgement is important - but it may be achieved through suitable combination of gateway URI, Logo or Text.m