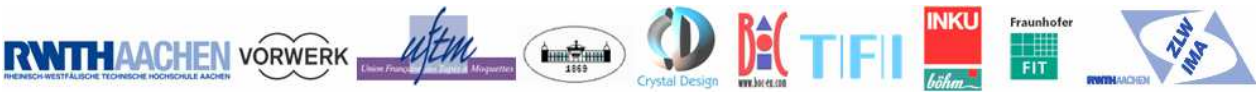
AsIsKnown
FP6-IST-4-28044



AsIsKnown

A semantic-based knowledge flow system for the European home textiles industry

Work package 3: Common sense ontology engineering

Deliverable D7 ”Report on Requirement Analysis (Ontologies)”

31.07.2006

Lead Editors:
Kiril Simov (IPP-BAS)
Petya Osenova (IPP-BAS)

Status: v1.0

Context

WP 3	Common Sense ontology engineering
Task 3.1	Definition of use cases as part of the requirements analysis. The use cases definition will be performed in the context of the semantic-based web services of AsIsKnown.
Dependencies	This deliverable requires user requirement input from WP 02, Task 2.1 and 2.2

Contributors: BOC, TFI, FIT, IPP-BAS, RWTH, CD

Approved by: Kiril Simov, Bulgaria as WP3 leader

Executive Summary

This report views the AsIsKnown architecture from the Ontology engineering point of view. The AsIsKnown Onto System (AIKOS) is understood as the 'lingua franca' component, through which all the modules within the system communicate between themselves and with the external world. Towards the creation of AIKOS several sub-tasks come in order. First of all, the explication of the 'communicative mapping points' in the overall architecture is to be done. Then, the upper and domain ontologies have to be built complying to two main approaches: bottom-up and top-down. After the creation of the ontology network, more operations need to be ensured over them. For example, editing, storage, inference etc. For that reason the tasks are divided into two parts: Ontology Use Cases and Ontology Related Services. The former handles the ontology design, ontology inference, communication and text analysis. The latter focuses on ontology dynamic maintenance.

Table of Content

List of Figures

1. Introduction and Problem Statement

In our work for ontology we adopt Tim Gruber's definition: "An ontology is a formal, explicit specification of a shared conceptualisation." [1] We consider ontology as a set of classes (concepts) which are interconnected via properties (relations). The set of relations includes sub-concepts (is-a, kind-of), part-of, caused-by, used-for, etc. In order to be more concrete we also select OWL as an ontology representation language [2].

In this deliverable we discuss the requirements on usage of ontologies within the framework of AsIsKnown project with respect to the following points:

1. Interaction with the other components of the overall AsIsKnown Architecture
2. Definition of ontological classes and properties with respect to the need of communication within the area of home textile industry.

The first point ensures the appropriate level of interoperability within the AsIsKnown Architecture, and the interaction between the components of the architecture and the external world. Here our task is to identify the necessary services to be implemented within the AsIsKnown Onto System (AIKOS). These services have to provide support of the whole range of the usage of the ontology within the AsIsKnown architecture. These include the creation of the ontology and extension of the ontology during its usage; exploring the represented classes and properties to fulfil the tasks of the other components of the architecture; interaction to external world (other content representation systems, like relational databases, ontologies, etc, and human users.

The second point is connected to the approach for construction of the common sense ontology in the area of the home textile industry. We envisage the usage of two complementary approaches to the ontology creation – bottom-up approach and top-down approach. The first one starts with standards and terminological dictionaries in the area of home textile. Having analysed these sources of domain information the approach generates a set of domain classes and properties. The second approach map these domains classes and properties to the classes and properties within an upper level ontology which ensures the conformity of the domain ontology to the ontological commitments implemented within the upper ontology and consistency of the domain ontology. Additionally, the bottom-up approach provides as a result lexicon(s) aligned to the domain ontology. Such lexicons are very useful for the interaction with human users and analysis of natural language texts.

Here we discuss the services that will be necessary to support the incorporation of the Ontology Management System within the overall AsIsKnown Architecture. We first discuss the current architecture of the AsIsKnown. Then we determine the roles of the ontology and the ontology management system with respect to the architecture. These roles are outlined in four use cases: *Ontology Design*, *Ontology Inference*, *Communication*, and *Text Analysing/Semantic Annotation*. Then we describe the services that are necessary for supporting these use cases. The use case of the creation of domain ontology will be supported with the same services as the ones described in this report. Thus the approach to the construction of the common sense ontology will be presented later in deliverable D6 "Report on the existing standards in home textiles industry domain as a domain ontology."

2. Ontology Management Architecture and Service Requirements

In this section we first represent a short overview of the current AsIsKnown Architecture. Then we discuss the role of the ontologies and the ontology management system within this architecture.

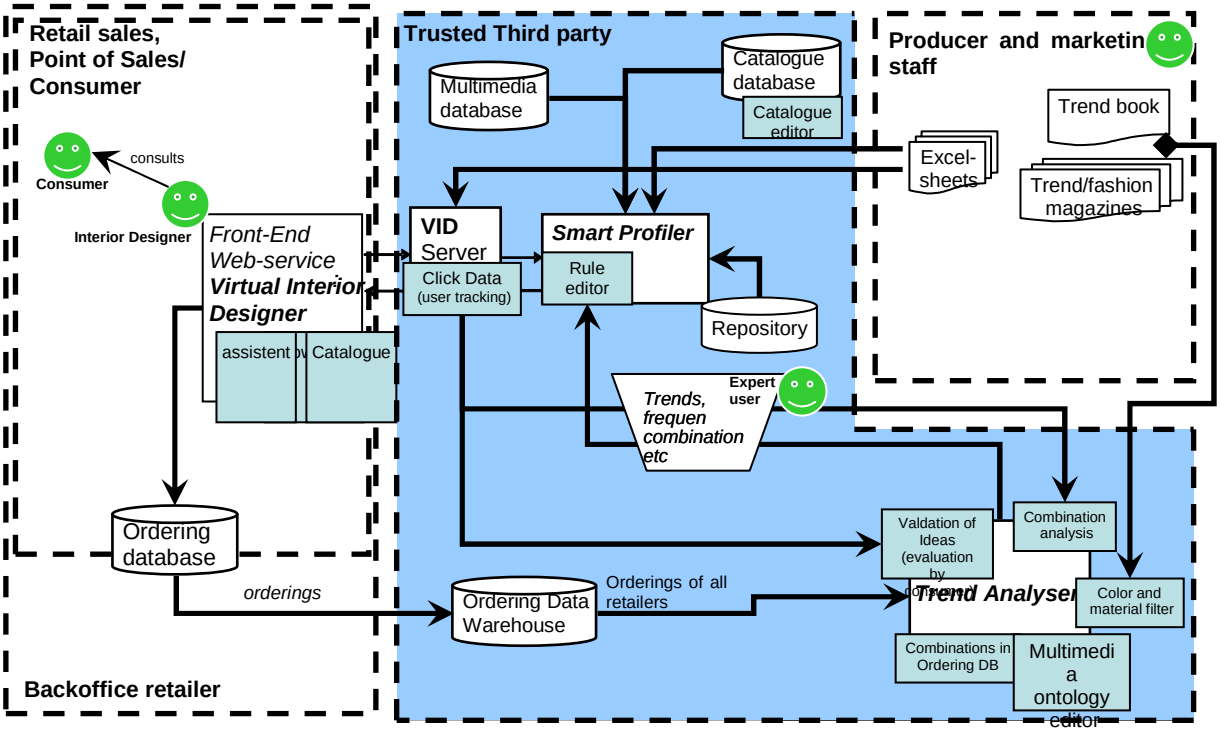


Figure 1: The AsIsKnown Architecture

The overall AsIsKnown Architecture is tripartite and it comprises Producer/marketing component, Retailer/Consumer component and AsIsKnown core component (depicted in blue and named as Trusted Third party). The first two components exist already in the current infrastructure supporting the European home textile industry. The aim of the AsIsKnown project is to complete the current infrastructure with services supporting better the communication between producers and customers in terms of actual transactions and prediction of the future market development. The AsIsKnown core component consists of two main services – *Smart Profiler* and *Trend Analyser*. These two modules interact with several internal and external repositories of information - *Catalogue database*, *Multimedia database*, *Ordering Data Warehouse*, *Repository*, *Producers databases*, *Ordering database*, *Trend books/magazines*, etc. The general aims of the two services are defined in the Annex I - "Description of Work" (page 28): "The Smart Profiler acts as a mediator between different knowledge bases and knowledge-intensive tasks within the business process by providing a common data format for design processes of home textiles." and (page 29) "The Trend Analyser is a tool suite to support marketing staff in the home textiles industry, which, together with the Smart Profiler, will integrate data across the industries' organisations. Trends and combinations are detected and evaluated by a human expert aided by text and data mining. The expert then formulates trends and combinations aided by the common sense ontology." Some of the information flows in the system are:

1. The information from the producers' databases is collected by the Smart Profiler into the Catalogue database and the Repository. Then it is provided to the Virtual Interior Designer. The Multimedia database is used as an additional source of information for the Virtual Interior Designer depending on the user profile and needs. In this process of information transfer the Smart Profiler needs to unify the vocabularies used in the producers' databases, the Catalogue database, the Multimedia database (the descriptors attached to the images), the Repository and the Virtual Interior Designer.
2. The Virtual Interior Designer transfer information back to the Smart Profiler in two formats: virtual interior designs and queries. The first type of information is used to define user profiles and to determine trends. The second type of information is used for searching of information in the Catalogue database, Multimedia database and the Repository. Again a need for unifying the vocabularies used in the different information sources is present.
3. The Trend Analyser receives information from the Ordering Data Warehouse (populated with data from the Ordering databases), the Virtual Interior Designer Server, the Trend books and magazines. Having processed the information in a semi-automatic way, the Trend Analyser generates trend rules which are applied over the data in the Repository, the Catalogue database and the Multimedia database for validation and description of trends. The process requires the usage of the same vocabulary among the different components in the AsIsKnown core component and unification with the external sources.

On the basis of these information flows we identify the role of the ontology and the ontology management system in the AsIsKnown core component as it is presented on Fig. 2

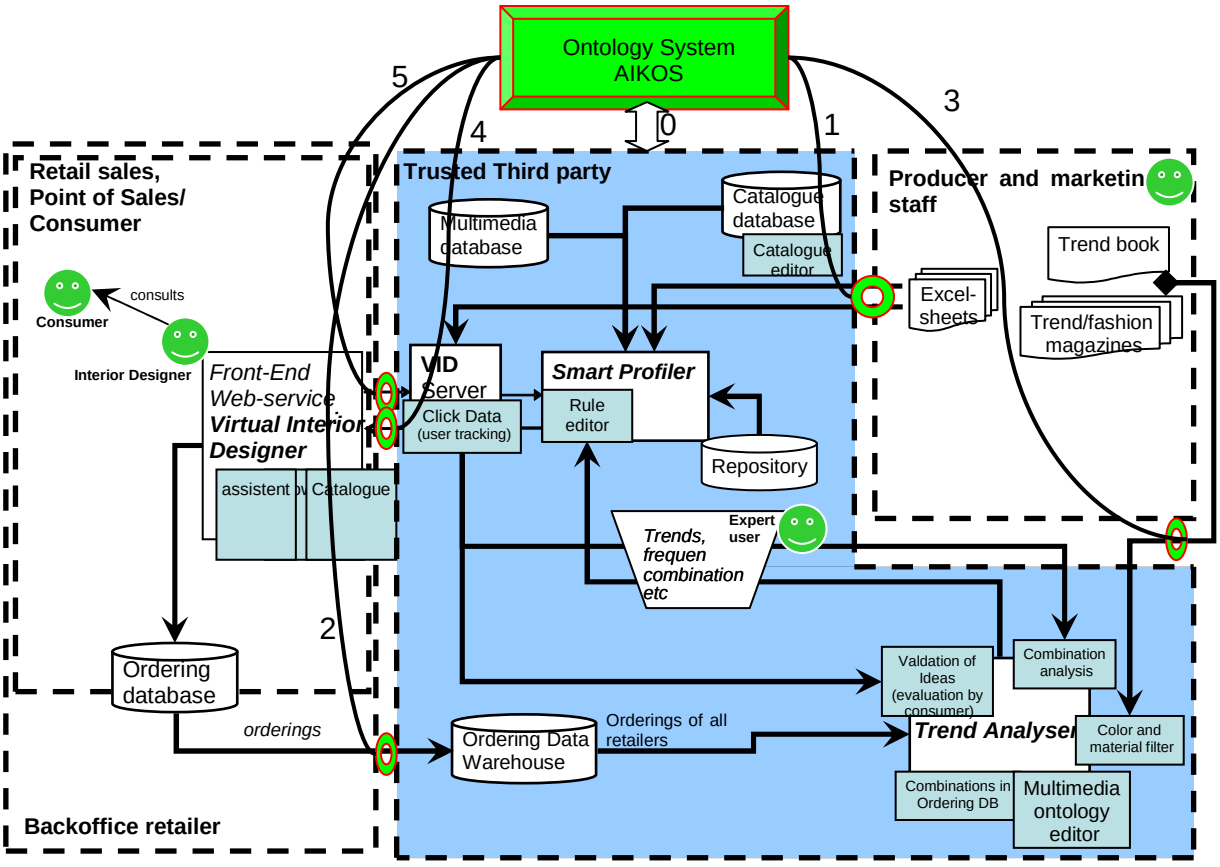


Figure 2: The AsIsKnown Onto System and its interaction with the rest of the architecture

The AsIsKnown Onto System (AIKOS) will provide the services necessary for supporting the information flows as they were described above. It will be integrated into the AsIsKnown Core Component. This integration is depicted in the Fig. 2 by the block arrow **0**. AIKOS will ensure the usage of a common vocabulary among the elements of the AsIsKnown Core Component, searching and inference over the ontologies, modularisation of the ontologies. Additionally, AIKOS will support the mapping of the ontologies within AsIsKnown Core Component to external content sources or users. These mappings are depicted by links **1** to **5**. AIKOS will provide the following services:

- Editing of Ontologies
- Storage of Ontologies
- Inference Services (Consistency, Realization, Classification, Navigation)
- Mapping of Ontologies and Content Systems (links 1,2,3,4,5)
- Semantic Annotation with Ontological Information (link 3)

2.1. Ontology Use Cases

In this section we discuss the use cases for the ontology management system within the AsIsKnown architecture. In parallel each of the above services are discussed in connection to a use case within the whole system. The use cases are as follows:

- *Ontology Design*. This use case covers the following activities: *initial creation of the ontology*; *modification of the ontology*; *modularisation of the ontology*. The initial creation of the ontology is related to the usage of the AsIsKnown System, but it is out of the typical usage of the system. This activity requires some of the services related to the use case, but there are also other services specific to the creation of domain ontology. We will consider a methodology for domain ontology creation below. The modification of the ontology is a necessary operation for tuning the ontology to the tasks of the system. Typical examples of such tasks are:

  1. extension of the Catalogue database with new records – in this case new concepts might be necessary if the ontology does not cover important conceptual distinctions made in the new records;
  2. generation of trend description which requires some new classes and properties that are local to the trend;
  3. addition of new lexical term in some natural language.

  The modification of the ontology has to be done by a knowledge engineer with the help of expert in the domain. When the modification is temporary or local with respect to a task or a description the modification could be done by the user of the corresponding tool. For example, when a modification of the ontology is necessary for the definition of a trend then it can be done by the Trend Analyser user. Modularisation of the ontology is necessary to support customisation of the ontology to a particular task or description. For example, we can have several ontologies developed for supporting different trend generation approaches or different user profiles. Such task or a description could require extension or modification of the ontology. Such extension/modification will be stored in different modules. Interdependency among

such modules can be established by import of one ontology into another. All these activities will be supported by services that are grouped under the names Editing Ontology Services and Storing Ontology Services.

- *Ontology Inference.* This use case covers the following activities: *navigation over the ontology*; *inference of new information from the ontology*. The navigation over the ontology is simple traversing over the classes and properties in the ontology. It includes such tasks like 'give me the classes that are immediate subclasses of a given class' or 'give me the classes that are part of a given class'. The inference of new information from the ontology is a task which provides mechanisms for explication of knowledge that is implicit in the ontology. Typical inference tasks are: consistency check of a classes description; classification of a new classes description with respect to a taxonomy; realization of an instance (finding the most specific classes in a taxonomy which describe the instance). In the AsIsKnown Architecture the navigation over the ontology will be used when a user is searching for particular information. For example, the architect who is using the Virtual Interior Designer is formulating a description of the textile product he/she needs for the current design. The way in which the architect is defining his description requires specification of a sub-concept of a given concept (for instance, sub-concept of the concept material). Thus, the formulation of the description requires navigation over the ontology. Then the description is translated in an ontological class description which is send to the Smart Profiler for evaluation over the Catalogue database and the Repository. The description is classified with respect to the taxonomy of the ontology and then all the instances of the classes that are under the new classified class description are an appropriate answer to the need of the architect. This use case will be supported by the services named Inference Services.

- *Communication.* This use case covers the following activities: *mapping between content systems*; *translating of natural language elements (words, phrases) into ontological classes and properties*. The mapping between content systems is the mechanism for transferring of information from the external world to AsIsKnown Core Component. On Fig. 2 this transferring of information is depicted by the links **1** and **2**. The Producers databases and the Ordering databases usually use their own conceptualization of the domain of the home textile. The mapping will be done once for each concrete conceptualization. The mapping will be based on rules that connect the elements of the external conceptualization to the classes and properties in the ontology. The transfer of the information will be based on these rules. The result will be an extension of the ontology with instances from the external database. The translating of natural language elements (words, phrases) into ontological classes and properties activity supports the communication between the AsIsKnown Core Component and some natural language. We can not expect that we will support complete translation from or to a natural language, thus only partial solutions will be implemented. On Fig. 2 this translation is depicted by links **3**, **4** and **5**. Link **3** represents the connection of free natural language text with the ontological information in order to support the Trend Analyser in the generation of trend description. This task is very complicated and it is defined as a separate use case. Link **4** ensures the mechanism in which a human user of the Virtual Interior Designer navigates over the ontology. Here we can not expect that the user will be able to use the ontology directly in its formal representation. To overcome this problem we will use the lexicons that will be aligned to the ontology as a mechanism for ontology navigation. The actual user interface will be implemented as a system of menus that allow the user to specify class's descriptions. Link **5** depends on the output of the Virtual Interior Designer. We envisage two kinds of information: queries about ontological data and ready (virtual) designs. The first kind of data will be class descriptions that have been

already generated by the system menus and they will be already defined in the vocabulary of the ontology and thus there will be no need for further translation. Depending of the design descriptions some of the approaches already discussed will be used in order with regard to the description to be translated into terms used inside of the AsIsKnown Core Component. The use case will be supported by the services defined in Mapping of Ontologies and Content Systems Services and Semantic Annotation with Ontological Information Services;

- *Text Analysing/Semantic Annotation.* This use case supports all the activities necessary for realizing of the link **3** – translation from free natural language texts into the ontology. There are two steps of processing necessary to ensure this mapping: text analysing and semantic annotation. The first step ensures the recognition of the relevant natural language elements to be annotated with ontological information. This step comprises at least the following processing: converting of documents in different formats (PDF, RTF, HTML, etc) into a common representation of the content (text and images), tokenization of the text, lexical analysis and partial parsing. The result of the text analysis will be used in the process of semantic annotation. The semantic annotation will be supported by the lexicons aligned with the ontology. This process will be semi-automatic. The user will be able to navigate over the result of the text analysis and to attach ontological information to it. The ontological information will be classes that define the meaning of the phrases in the text and properties which define some relations between entities mentioned in the documents. The results of the annotation will be searchable via the Trend Analyser.

2.2. Ontology Related Services

In this section we present a summary of the requirements for the services which the AIKOS minimally will have to provide. Here we provided the well established services of the description logics [3] and ontology management [4]. Where it is necessary we extend or modify these services.

Editing of Ontologies

- *Analytical Ontology Editing* – these services have to support the editing of the ontology directly in some serialization of the ontological language. For example, XML representation of OWL;
- *Visual Ontology Editing* – these services have to support visualization of a portion of an ontology and support editing via forms or via direct manipulation of the graphical representation;
- *Specific to AsIsKnown Editing* – these services have to support designing and editing of templates for reflecting the ontology structure;
- *Lexicon Editing* – these services have to support the creation of lexical entries and their alignment to the ontology.

The result of the editing services will be checked for validity and consistency (depending on the kind of the processed information), and then the ontology and the lexicon will be updated.

Storage of Ontologies

As it was mentioned above, the ontology will be represented in a modular way. Thus, we will use the term *repository* for the data structure which contains one module of the ontology. Each module will have a name and it will be possible to import modules into other modules. Thus, the services that will be supported are:

- *Repository Management* – these services have to support the creation of a new repository, deletion, renaming, coping of an existing one. When modifications are carried out, this will have an impact on other repositories and appropriate messages will be presented to the user;
- *Ontology Management* – these services have to support the manipulation of the ontological information within one repository – the standard operations of adding, deleting and updating on the level of class, property and instance descriptions. Corresponding checks for validity and consistency will be performed;
- *Modularization* – the services of this group have to support the interaction between repositories via the import operators.

Inference Services

- *Navigation* – these services have to support traversing of explicit information represented in the ontology;
- *Consistency* – these services have to support the check for contradiction within one repository. At least the consistency of the whole ontology and the consistency of a class description with respect to an ontology have to be supported;
- *Classification* – this service has to support for a class ***C*** the finding of the minimal classes in the ontology that are more general than ***C*** and the maximal classes in the ontology that are more specific than ***C***.
- *Realization* – this service has to support for an instance ***I*** the finding of the minimal classes in the ontology that describe ***I***.
- *Instance Checking* – this service has to support for an instance ***I*** and a class ***C*** the checking whether ***I*** is an instance of ***C***.
- *Retrieval* – this service has to support for a class ***C*** the finding of all of its instances in the ontology.
- *Rules* – this service has to support the creation of user defined rules. User-defined rules are a very powerful mechanism for changing the ontology. There will be a mechanism for defining the scope of such rules to one or few repositories.

Mapping of Ontologies and Content Systems

- *Mapping Creation* – this service has to support the creation of mapping rules from an external actor with respect to the AsIsKnown Core Component conceptualization to the AsIsKnown ontology. The rules will be determined on the basis of the classes and properties within the ontology. If there are missing classes or properties they could be added to the ontology with the Editing services. The result of the application of a mapping rule has to be valid (consistent) OWL statements about some instances;
- *Mapping Rule Application* – this service has to support the application of the mapping rules from a given mapping and storing of the resulting OWL statements in a given repository. Reaction of inconsistency or other problems will be supported by the Storage of Ontologies services;

- *Lexicon-based Mapping* – this service has to support the mapping from the ontology to a system of menus and back. The items of the menus are defined with the help of the lexicons aligned to the ontology.

Semantic Annotation with Ontological Information

- *Text Analysing* – these services have to support the whole range of processing of the multimedia documents from converting them from original format into a common (XML-based) format to partial parsing (if necessary). We cannot provide such services for all languages (English, German, French, Bulgarian), but we will provide the basic tools for supporting such services and we will provide some for Bulgarian and English.
- *Semantic Annotation* – these services have to support the annotation with ontological information of the already processed text. Minimally, the services have to provide mechanisms for annotation of phrases with ontological classes and for annotation of relations between entities.

3. Conclusions and Outlook

In this report we analysed the requirements to the ontology management system to be implemented within the AsIsKnown system. First, we have identified the roles of the ontology within the overall AsIsKnown architecture. Then we defined the appropriate use cases for the ontology management system. For each use case examples of activities that are necessary to be supported and which services of the ontology management system will support these activities are given. Then we presented a list with minimal set of required services. These services will the basis for the design and the implementation of the AsIsKnown Onto System. The implementation will be based on available ontology management tools as far as possible. The text processing and ontology annotation will be implemented within the CLaRK System developed at IPP-BAS [5].

4. References

[1] Gruber, T.R. (1993). *A translation approach to portable ontology specifications*. Knowledge Acquisition, 5, pp. 199-220.

[2] OWL. *Web Ontology Language (Overview)*. http://www.w3.org/TR/owl-features/

[3] Sean Bechhofer, Ian Horrocks, Peter F. Patel-Schneider, Sergio Tessaris. A Proposal for a Description Logic Interface. In: (edited by: Patrick Lambrix, Alex Borgida, Maurizio Lenzerini, Ralf Mueller, Peter Patel-Schneider) Proceedings of the Intl. Workshop DL'99, Linköping, Sweden, July 30 - August 1, 1999.

[4] Atanas Kiryakov, Kiril Simov, Damyan Ognyanov. Ontology Middleware and Reasoning. In the "Towards the Semantic Web: Ontology-Driven Knowledge Management", editors John Davies, Dieter Fensel, Frank van Harmelen. John Wiley & Sons, Europe, 2002.

[5] Kiril Simov, Zdravko Peev, Milen Kouylekov, Alexander Simov, Marin Dimitrov, Atanas Kiryakov. *CLaRK - an XML Based System for Corpora Development*. UCREL Technical Paper number 13. Special issue. Proceedings of the Corpus Linguistics 2001 conference, edited by Paul Rayson, Andrew Wilson, Tony McEnery, Andrew Hardie and Shereen Khoja. ISBN 1 86220 107 2. Lancaster University (UK), 29 March - 2 April 2001

A. Appendix: List of Abbreviations

AIKOS	AsIsKnown Onto System
HTML	Hyper-Text Markup Language
OWL	Web Ontology Language
PDF	Adobe Portable Document Format
RTF	Rech Text Format
XML	Extensible Markup Language