EU-IST Project IST-2004-026460 TAO
TAO: Transitioning Applications to Ontologies

D1.3.2 Case study reports on evaluating the methodology

Tomás Pariente (Atos Origin S.A.E.)
Germán Herrero (Atos Origin S.A.E.)
Danica Damljanovic (University of Sheffield)
Farid Cerbah (Dassault Aviation)

Abstract

EU-IST Specific targeted research project (STREP) IST-2004-026460 TAO
Deliverable D1.3.2 (WP 1)

This report presents the evaluation of the TAO methodology for transitioning the TAO case studies into semantics-based SOA systems. In order to evaluate software, methodologies and products, some kind of criteria or metrics have to be taken into account. Hence, this report presents a brief study of the most relevant metrics used in evaluation processes. In particular usability and technical metrics are considered suitable for the evaluation of the TAO methodology in the context of TAO case studies. Due to the nature of these metrics, the evaluation is also applied to the TAO tools and in particular to the TAO Suite.

Keyword list: evaluation, usability metrics, technical metrics, legacy systems, methodology, transitioning process, legacy system, semantics-based SOA, ontologies.

WP1 SWS Bootstrapping methodology
Nature: Report
Contractual date of delivery: 28/02/2009
Reviewed By: Zlatina Marinova

Document ID: TAO/2008/D1.3/v2.0
Dissemination: PU
Actual date of delivery: 14/04/2009

TAO Consortium

This document is part of a research project partially funded by the IST Programme of the Commission of the European Communities as project number IST-2004-026460.

University of Sheffield
Department of Computer Science
Regent Court, 211 Portobello St.
Sheffield S1 4DP
UK
Tel: +44 114 222 1930
Fax: +44 114 222 1810
Contact person: Kalina Bontcheva
E-mail: K.Bontcheva@dcs.shef.ac.uk

University of Southampton
Southampton SO17 1BJ
UK
Tel: +44 23 8059 8343
Fax: +44 23 8059 2865
Contact person: Terry Payne
E-mail: trp@ecs.soton.ac.uk

Atos Origin Sociedad Anonima Espanola
Dept Research and Innovation
Atos Origin Spain, C/Albarracin, 25, 28037
Madrid
Spain
Tel: +34 91 214 8835
Fax: +34 91 754 3252
Contact person: Tomás Pariente Lobo
E-mail: tomas.parientelobo@atosresearch.eu

Jozef Stefan Institute
Department of Knowledge Technologies
Jamova 39
1000 Ljubljana
Slovenia
Tel: +386 1 477 3778
Fax: +386 1 477 3131
Contact person: Marko Grobelnik
E-mail: Marko.Grobelnik@ijs.si

Mondeca
3, cité Nollez
75018 Paris
France
Tel: +33 (0) 1 44 92 35 03
Fax: +33 (0) 1 44 92 02 59
Contact person: Jean Delahousse
E-mail: jean.delahousse@mondeca.com

Sirma Group Corp., Ontotext Lab
Office Express IT Centre, 5th Floor
135 Tsarigradsko Shosse Blvd.
Sofia 1784
Bulgaria
Tel: +359 2 9768 303
Fax: +359 2 9768 311
Contact person: Atanas Kiryakov
E-mail: naso@sirma.bg

Dassault Aviation SA
DGT/DPR
78, quai Marcel Dassault
92552 Saint-Cloud
Cedex 300
France
Tel: +33 1 47 11 53 00
Fax: +33 1 47 11 53 65
Contact person: Farid Cerbah
E-mail: Farid.Cerbah@dassault-aviation.com

Executive Summary

This report presents the final evaluation of the TAO methodology in the context of TAO case studies. Taking into account the objectives and scope of TAO, the methodology defines the theoretical and conceptual steps in the transitioning process necessary to migrate legacy systems into semantic systems, especially into semantics-based SOA systems. In particular, this report evaluates the TAO methodology when it is applied for transitioning the TAO case studies into semantics-based SOA systems.

During a general evaluation process of software, methodologies and products, some kind of criteria or metrics need to be taken into account. Hence, this report presents a brief study of most relevant metrics used in evaluation processes. Consequently, the evaluation of TAO methodology presented in this report should be based on some metrics. In particular, in order to evaluate the TAO methodology, usability and technical metrics will be taken into account. However, due to the nature of these metrics, the evaluation will not only be applied to the theoretical process (i.e. the methodology) but it will be also extended to the software tools. Consequently, metrics will be applied to TAO Suite or TAO components, as a complement in the evaluation of TAO methodology.

Hence, this report generally tries to evaluate the overall TAO transitioning process which is theoretically defined by TAO methodology and technologically supported by TAO tools, such as TAO Suite or TAO components.

Acronyms

API	Application Programming Interface
DASSAV	“Dassault Aviation” partner
DoW	Description of Work
GUI	Graphic User Interface
IEC	International Electro technical Commission
IDE	Integrated Development Environment
ISO	International Organization for Standardization
IT	Information Technology
JSI	“Jozef Stefan Institute” partner
MON	“Mondeca” partner
ONTO	“Ontotext Lab, Sirma Group Corp” partner
OWL	Web Ontology Language
RCP	Rich Client Platform
RDF	Resource Description Framework
RTD	Research and Technological Development
SA-WSDL	Semantic Annotations for WSDL Working Group
SOA	Service Oriented Architecture
SWS	Semantic Web Services
UI	User Interface
UML	Unified Modeling Language
USFD	“University of Sheffield” partner
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
XML	eXtensible Mark-up Language

Contents

1 Introduction

The objective of this report is the evaluation of the TAO methodology applied to transitioning the TAO case studies into semantics-based SOA systems. This report is an update of the previous evaluation [5] based on an earlier version of the methodology [2]. Figure 1-1 depicts the dependencies between the versions of the aforementioned reports.

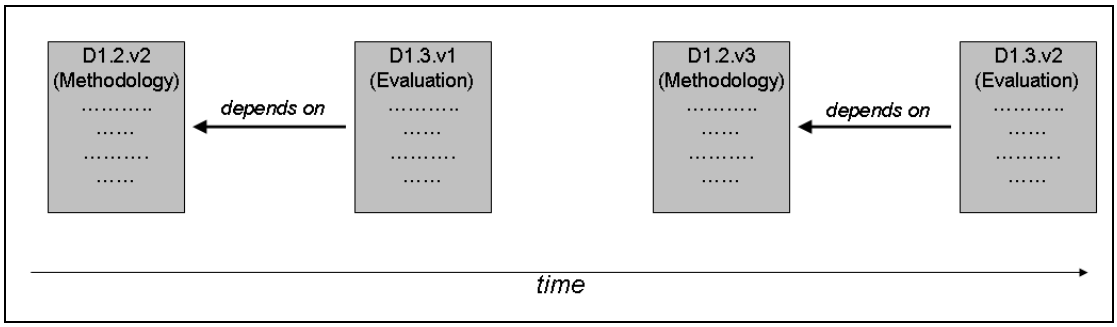

Figure 1-1 Dependencies between D1.2 and D1.3 reports

As in the previous version, an explanation of the terminology used in this report can be found in appendix 6.1. Additionally, due to nature of the metrics used in this report, some aspects of the evaluation are actually applied to the TAO tools and TAO Suite, rather than to the conceptual methodology.

The organization of this report is as follows:

- Section 1 introduces the report focus, describing the scope of the two versions of this report.
- Section 2 explains the motivation for carrying out an evaluation of the TAO methodology taking into account its relevance with respect to TAO objectives and other work packages.
- Section 3 presents a brief state of the art of metrics that can be used to carry out the evaluation process.
- Section 4, after summarizing briefly the TAO methodology, describes the evaluation of TAO methodology applied to the TAO case studies. Usability and technical metrics will be used.
- Section 5 suggests some improvements for the TAO methodology, taking into account the evaluation results.
- Finally, section 6 concludes the report with a brief summary.

2 Relevance to TAO

This section explains the relevance of this deliverable to the TAO project. Specifically, section 2.1 explains the relevance of this deliverable to TAO objectives, hence providing the motivation of this deliverable, and section 2.2 describes the relevance of this deliverable to other TAO work packages.

2.1 Relevance to project objectives

According to the DoW, there are four types of objectives, which are explained in the next sub-sections.

2.1.1 Overall goal

Basically, the overall goal of TAO is to define a low-cost route for the transition of legacy system to semantic-based SOA systems. Starting from existing research and technology in different areas and enhancing them, TAO aim is building an infrastructure for the transition of legacy systems.

This deliverable evaluates the TAO methodology applying suitable metrics, mostly qualitative, and using the experience gained in the case studies.

2.1.2 Innovation

The main innovation of TAO is semantic Web-service bootstrapping – an innovative methodology based on state-of-the-art ontology learning and semantic data integration.

This objective is related with WP1 since it defines the aforementioned innovative methodology. This deliverable is directly related to the innovation objective since it tries to measure and evaluate the innovative level of the bootstrapping methodology.

2.1.3 Research Objectives

The process of transforming monolithic, legacy applications into semantic SOA requires new research in several areas, which constitute the three RTD objectives of TAO:

- SWS bootstrapping via semi-automatic acquisition of domain ontologies
- Augmentation and integration of legacy content
- Transitioning Methodology and Infrastructure

This deliverable is obviously directly related to the third sub-objective, but it is also relevant for the remaining sub-objectives, since it evaluates also the software infrastructure (i.e. TAO Suite, and tools) necessary for achieving them.

2.1.4 Exploitable Results

This is the second version of the evaluation of the TAO methodology from the point of view of the case studies. The first version of the document [5] was used to give feedback to the last version of the TAO methodology, which is under evaluation in the present report.

Although this deliverable does not intend to drive a new version of the methodology, it is expected that it helps to the uptake of the methodology from an exploitation viewpoint by checking the suitability of the methodological approach for transitioning from legacy to ontology-based applications.

2.2 Relation to other work packages

This deliverable evaluates activities and sub-activities of the TAO methodology which are supported by techniques and components defined in WP2, WP3 and WP4.

Furthermore, this deliverable evaluates whether the TAO methodology is aligned with the functionalities of the software tool provided by WP5 (TAO Suite), and to what extent the TAO methodology can be supported by the TAO Suite or by the components integrated into it.

The document is using input from the case studies (WP6 and WP7), since the TAO methodology is evaluated in the light of its application to industrial cases.

3 Metrics for evaluating TAO methodology: State of the art

In order to evaluate the TAO methodology, first of all it is necessary to revisit the criteria or metrics used for this evaluation. According to the DoW, the evaluation of TAO methodology should be based on usability and technical metrics.

These metrics belong to the field of software metrics, because software metrics are not only useful for evaluating software products (e.g. software tools), but also other artefacts belonging to IT field, such as software entities in general. Hence, these metrics can be used for evaluating the TAO methodology where the TAO methodology is considered as a kind of software entity. Consequently, a brief state of the art about software metrics is presented in this section. Afterwards, the usability metrics and technical metrics are explained in order to evaluate the TAO methodology.

3.1 Software metrics: state of the art

Metrics are an integral part of the state-of-the-practice in software engineering. Nowadays, customers are specifying and using metrics to better understand, track, control and predict software entities, such as software products, resources and processes. Software measurements (i.e. software metrics) are used for collecting qualitative data about software entities and their processes in order to increase their quality.

The usage of measurement is common not only in software engineering. Measurements are used in everyday life to do such things as check the time we need to go to work or the food dosage we need at lunch time. Measurements are used extensively in most areas of production and manufacturing to estimate costs, calibrate equipment and assess quality. Engineering disciplines depend on the rigor that measurements provide, but what does measurement really mean?

The term “software metric” has different meanings depending on the source: the meaning varies from project cost and effort estimation to specific test performance measurement. One of the most used definitions of “software metric” is the following: "The continuous application of measurement-based techniques to the software development process and its products to supply meaningful and timely management information, together with the use of those techniques to improve that process and its products." [1]. Figure 3-1, illustrates the software metric definition to include software-related services.

Figure 3-1 Software metrics

Software metrics can provide the information needed by engineers for technical decisions and evaluations as well as information required by management. It is clear that a metric provides useful information, so that the selection and design phase of metrics for a product are significant for a successful evaluation and for obtaining relevant information that could improve the product.

The needs of a measure process are various; from weighing the performance of the entity to the evaluation of the improvement activities for the entity. For a better understanding of how to apply software metrics, the engineer should understand the meaning of the metrics and why he needs them. These metrics are not absolute but objective, and provide a framework to evaluate the quality of a software entity based on a set of rules or measures.

Also, "software metric" is a term applied widely to different activities such as:

- Metrics and models for cost and effort estimations, productivity metrics and models
- Quality control
- Data collection
- Quality metrics and models,
- Reliability models,
- Algorithm complexity

Consequently, the TAO methodology can be considered as a software entity and its evaluation can be based on software metrics.

Taking into account that software metrics improve the quality of software entities, we can identify three different types of software entities which are measureable:

- Processes: Software activities that involve time factors. Some interesting attributes of these activities that can be measured are: time (i.e. duration of the process), effort (i.e. associated to the process) or number of incidents occurred in the process (i.e. number of requirements errors detected in the development).
- Products: Products include deliverables, devices, tools, artefacts and documents generated along the software lifecycle. Some interesting external attributes are measurable, such as code reliability, understanding of a technical document, source code maintenance, etc. Other internal attributes are: length, functionality, modularity, etc.
- Resources: Entities which are input of the software production, such as human resources, materials, or methodologies. In this case, it is possible to measure to some extent cost and productivity of the resource.

The main objective of software metrics is the evaluation of a software entity. In our case this software entity is the TAO methodology. The reason of this is based on the fact that this software entity is not the typical software component of a framework

(i.e. a software module for user management in any application, an IDE Eclipse RCP application, etc.). TAO methodology is a software entity, because it can be considered as a software resource which can be measurable, according to the classification shown in Figure 3-2.

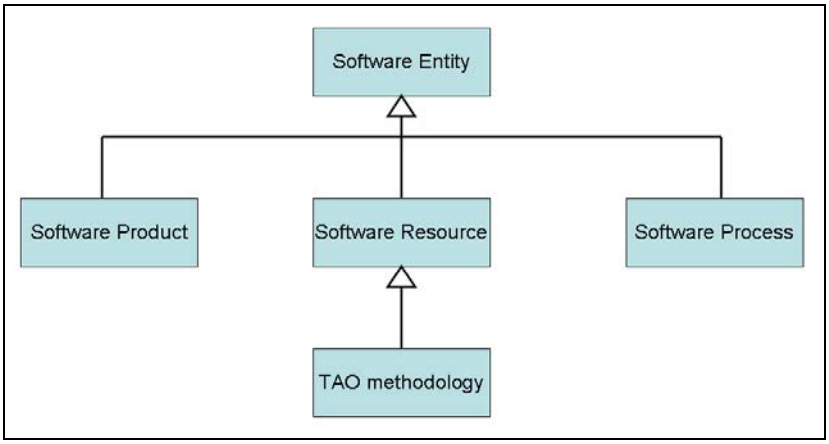

Figure 3-2 Measurable software entities

However, in the actual scene of software engineering, some sources point out that software metrics are controversial, because they cannot be deployed, and are not very useful since most of them cannot be applied in a typical software application lifecycle.

As is described before, the software metrics should have concrete goals, which determine the entities and the attributes that should be measure. But these goals change according to the type of user involved in the evaluation: for example managers, engineers and end-users have different point of view.

Consequently, in order to apply the software metrics for evaluating the TAO methodology, different measurable goals have to be set up and selected. And in fact, quality models (e.g. ISO 9126) are responsible for defining the software metrics by setting up and selecting the measurable goals in a standard way.

However, although developing and deploying a quality model is quite complex, these models suggest splitting into smaller modules, characteristics and sub-characteristics that could be enumerated in a quality checklist. Hence, quality models include a set of characteristics which can be considered as metrics if the fulfilment of these characteristics is measurable.

But how can an organization use a quality model for establishing software metrics in order to evaluate a software entity? The most common way of doing this, is to establish a set of questions for each characteristic of the quality model in order to measure if the quality model characteristics are fulfilled by the software entity to be evaluated.

Figure 3-3 depicts this methodology: we see that software metrics evaluate software entities in order to improve their quality, and that software metrics are established by quality models.

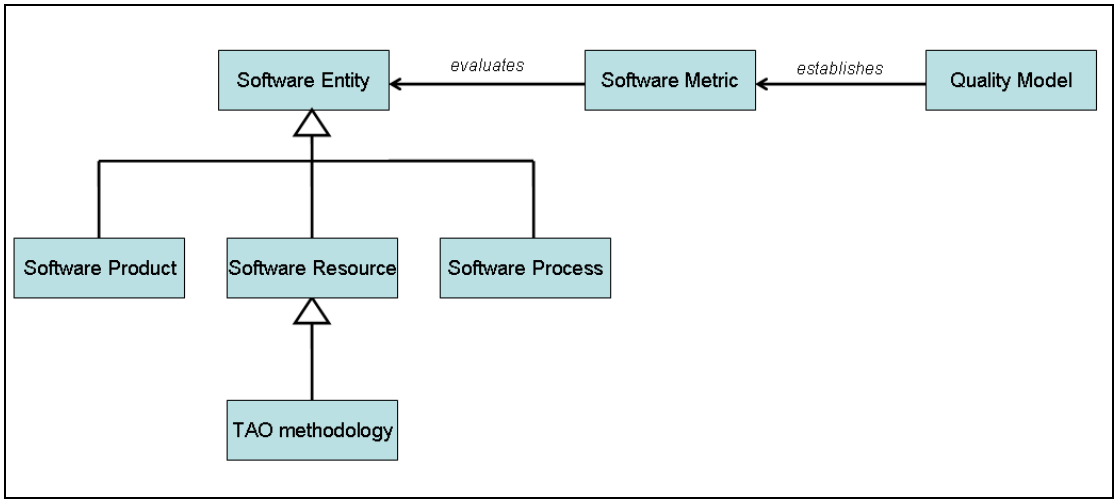

Figure 3-3 Software metrics and quality models

Consequently, in order to be compliant with standards, this report is based on the usability metrics and technical metrics established by the quality models.

A brief state of the art about quality models for software entities is presented in the following section.

3.2 Quality Models: state of the art

In the evaluation of software entities, the following quality models are the most commonly used.

3.2.1 ISO 9126

This quality model is an international standard for the evaluation of software quality provided by ISO (International Organization for Standardization). This standard is provided to developers, software quality reviewers and independent responsible persons who are in charge of specifying and evaluating the quality of a software entity.

This model is useful to check the completeness of a definition and identification of quality requirements, design goals, test and criterions to ensure the quality of the software product. The quality in any process of the software lifecycle (standard ISO 12.207) has influence on the quality of the software entity, and also, contributes to improve the quality of the product usability.

According to this standard, we can evaluate the quality of software if we measure internal attributes (i.e. static measures) and external attributes (i.e. behaviour of the software entity when it is used). In addition this standard has two other parts involved: a quality model and a quality in the usage of metrics.

The quality model proposed in this standard, classifies software entities according to a set of characteristics and sub-characteristics that must be evaluated. Figure 3-4 depicts this set of characteristics, and each of them is explained below:

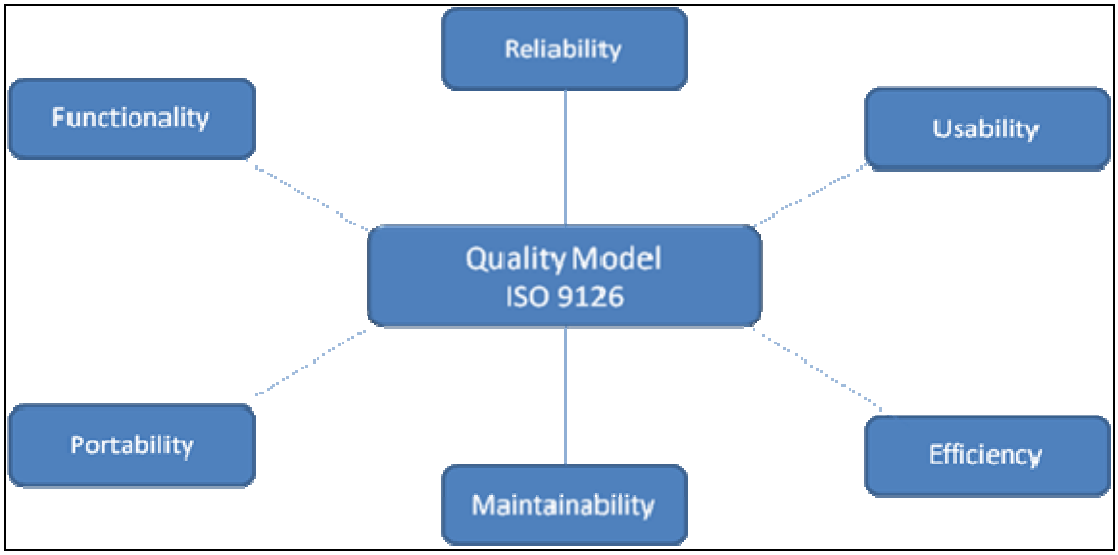

Figure 3-4 ISO 9126 Quality Model

- Functionality: software entity capability which provides a set of functions that satisfy the specific and implied requirements. The set of sub-characteristics are:
  - Suitability
  - Accuracy
  - Interoperability/Technological support
  - Compliance/Conformance
  - Security
- Reliability: software entity capability which provides maintenance for a level of performance under stated conditions and a stated period of time. The set of sub-characteristics are:
  - Maturity
  - Recoverability
  - Fault Tolerance
  - Conformance
- Usability: the capability that allows the end-user to learn, utilize, and understand the software entity. The set of sub-characteristics are:
  - Learnability
  - Understandability
  - Operability
  - Conformance
  - Attraction
- Efficiency: software entity capability that measures the performance of the software and the amount of resources used. The set of sub-characteristics are:
  - Time Behaviour
  - Resource Behaviour
  - Conformance
- Maintainability: software entity capability for changes and modifications. These changes include corrections, improvements, software adaptability, etc. The set of sub-characteristics are:
  - Stability
  - Analyzability
  - Changeability

    - Testability
    - Conformance
- Portability: software entity capability that supports the transference from one environment to other. The set of sub-characteristics are:
    - Installability
    - Replaceability
    - Adaptability
    - Conformance

This standard is used by some organizations to define the quality model for their software entities. However not all characteristics and sub-characteristics are used, so that some organizations define their specific quality model.

This quality model defines the software metrics as the measure of fulfilment of the characteristics explained above, by means of customized questions in the context of the software entity to be evaluated.

Taking into account that several of these metrics are applied when the software entity is running (e.g. a software tool), it seems to be difficult to apply these metrics to TAO methodology because they are conceptual guidelines and we cannot measure with performance metrics the execution of the methodology at runtime in any legacy system transitioning. Hence, these metrics should be applied not only to TAO methodology but also to the overall TAO transitioning process taking into account the TAO tools, such as TAO Suite[1]

3.2.2 ISO/IEC 15504

ISO/IEC 15504 is an international standard for software process improvement and evaluation and capability determination. This standard is based on SPICE, which is a framework for assessment of software products, based on a set of capability measures for all processes involved in the software lifecycle. This standard is a result of a joint technical subcommittee between ISO and IEC (International Electrotechnical Commission). The main outstanding result is a parallel process for the empirical evaluation of the results.

This standard defines a set of categories in which the assessors can place the evidence collected during the assessment. A weak point of this standard is that the capability dimension of the model has reached a high level of complexity and overlaps with the process dimension. So, the complexity of the evaluation and its cost is higher than for other quality or evaluation models.

3.2.3 CMMI

CMMI (Capability Maturity Model Integration) is a process improvement approach that provides organizations with the essential elements of effective processes. It can be used to guide process improvement across a project, a division, or an entire organization. CMMI helps integrate traditionally separate organizational functions,

[1] The functionalities of TAO Suite defined in D5.2 [5]

sets process improvement goals and priorities, provides guidance for quality processes, and is a point of reference for appraising current processes.

CMMI shows the path to reach a maturity level in an organization or a capability and quality level in a process. CMMI provides guidelines about what to do but not how to make it possible. CMMI is a broader version of CMM, on which it is based and from which it borrows most of its concepts, and offers best practices benchmarks for software development.

CMMI identifies a core set of software engineering process areas as:

- Requirements development
- Requirements management
- Technical solution
- Product integration
- Verification
- Validation

CMMI also covers other process areas, such as Process management, Project management and support.

Based on this model, we can define a set of metrics for the different process areas of the software engineering lifecycle and test the effectiveness. The metric has a key role in the verification and validation process. In the verification process the metric should ensure that the selected final products meet their specified requirements and the metric for process validation confirm that the final product, as provided, will fulfil its intended use.

3.2.4 Other approaches also based on ISO 9126

This sub-section describes other quality models based on ISO 9126. As explained before, this standard defines a set of quality characteristics that are analyzed thoroughly into sub-characteristics which are split in different attributes. The value of these attributes is estimated based on different performance measurements.

This hierarchical model is adopted in different domains. Bertoa and Vallecillo [18] develop a quality model based on ISO 9126 for software components COTS. Simao and Belchior provided and extended version of the standard in which they identify 124 quality parameters for software components.

Another quality model that is an extension of ISO 9126 is QUINT2 [10]. However this model is more oriented to the evaluation of software architecture quality. Franc and Carvallo [11] provide an ISO 9126 quality model adaptation for e-mail. Also, Botella [12] proposes a model for ERP selection using ISO 9126 as framework. Cai [13] suggests a quality model for components and systems based on components. Also, Fernandez and Rossi [14] define a quality model for distributed software.

The quality model proposed by Zo and Ramamurhty [15] is used to evaluate e-commerce websites in a B2C environment. Related to this domain, Webb and Webb [17] present different quality attributes and factors in websites that are relevant for end-users. Also, Parasuraman [16] describes SERVQUAL, a quality model defined by

five dimensions and 22 items that measure the different elements and quality characteristics of a service. The main idea of this model is the adaptation of the model to different environments depending on the functionality of the services in each environment and adapting the different dimensions described in the original model.

3.2.5 Conclusion

In the previous sections, we have mentioned that ISO 9126 quality model is the most used by organizations in order to establish software metrics for evaluating and improving the quality of their software entities.

In our case, ISO 9126 quality model provides the model and characteristics we should evaluate in the context of TAO, mainly the usability dimension (understandability, conformance, etc.) and the functionality dimension (interoperability, compliance, etc.). Additionally, the DoW requires evaluating the TAO methodology according to usability metrics and technical metrics, which belong to ISO 9126 quality model.

For the usability challenges in any software project, metrics evaluate the quality objectives and savings reached by the product (in terms of time and cost) and evaluate the technological impact of the software entity. In the context of TAO, usability metrics fits better for evaluating the TAO methodology.

For the functionality challenges (i.e. technical challenges) in software projects, software metrics provide us a better understanding of the technical process used in the product development and the product itself. At this level, we can measure the interoperability with other service architectures or the level of support in integrating services within third party workflows. Other sub-characteristics suggested by ISO 9126 at this level should be taken into account, such as suitability or compliance, for both case studies.

4 Evaluation of TAO methodology

4.1 Recalling TAO methodology

This section recalls the TAO transitioning process considering TAO methodology defined in D1.2.v3 [3].

A general methodology based on software engineering principles is proposed in [3]. The TAO transitioning methodology is one of the key scientific contributions expected from the TAO project. This TAO methodology is presented as a composite lifecycle, which highlights the interactions between existing methodologies for Service-Oriented Architectures and for ontology design. There is no single methodology for either of these tasks, rather, the TAO methodology takes the abstracted lifecycle sketches and demonstrates how and where these should be linked.

In short, the TAO methodology provides a refinement of the processes in the existing methodologies, and involves three key points of alignment between the ontology design process and the SOA design process:

- Service Oriented Ontology Learning: Learning ontologies from legacy data (e.g. source code, API documentation, database schema)
- Semantic Service Annotation: Using domain ontologies to annotate the provided services
- Service-Driven Ontology Refinement: Using feedback from service evaluation to refine ontologies.

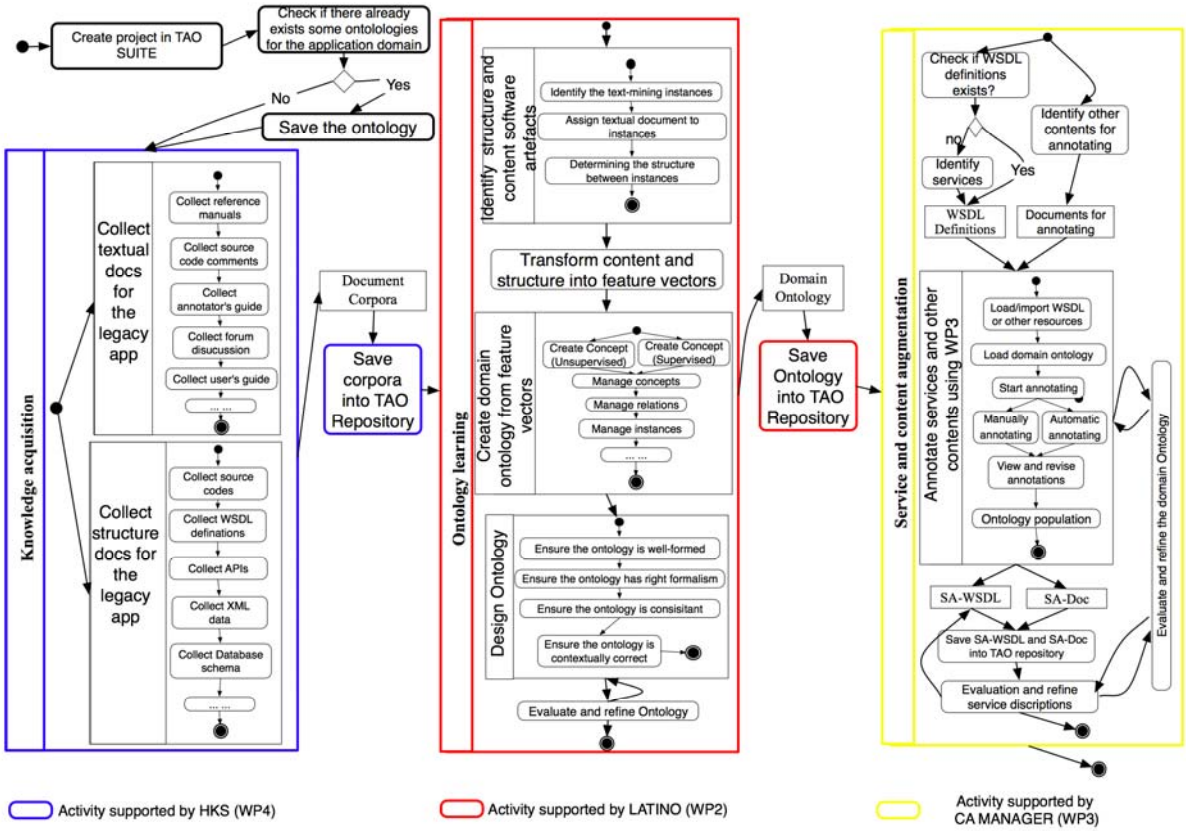

Figure 4-1 Cookbook TAO methodology [3]

Figure 4-1 shows the diagram depicting the final version of the TAO Methodology. In this figure the different phases (i.e. activities) involved in the transitioning process and their alignment with the key points highlighted before can be identified:

- Knowledge acquisition
- Ontology learning
- Semantic augmentation of legacy content and services.

The TAO methodology presents a cookbook-style guide on how the TAO suite can be used to assist the transitioning of a legacy application toward a semantic web platform. So, our objective in this deliverable is the evaluation of this methodology from the point of view of usability and functionality using the case studies as experience.

For these purposes, in the previous section we presented some literature about evaluation methodologies and a set of metrics that will be useful for evaluating the efficiency and functionality of the methodology. This evaluation of the methodology is carried out in the context of the case studies, thanks to the experience gained in TAO methodology and tools. The GATE and DASSAV case studies provided us with feedback that influenced in latests version of the TAO methodology.

4.2 Metrics from ISO 9126 to evaluate TAO methodology

We follow the same metrics established in the previous version of this deliverable [5]. This section is a reminder of the metrics selected in order to evaluate the TAO methodology.

In order to completely understand the metrics used for evaluation of TAO methodology, Figure 3-3 (from previous section) is refined including the fact that the software metrics are composed of a set of questions about the desirable software entity characteristics established by a quality model (see the software entity characteristics explained in section 3.2).

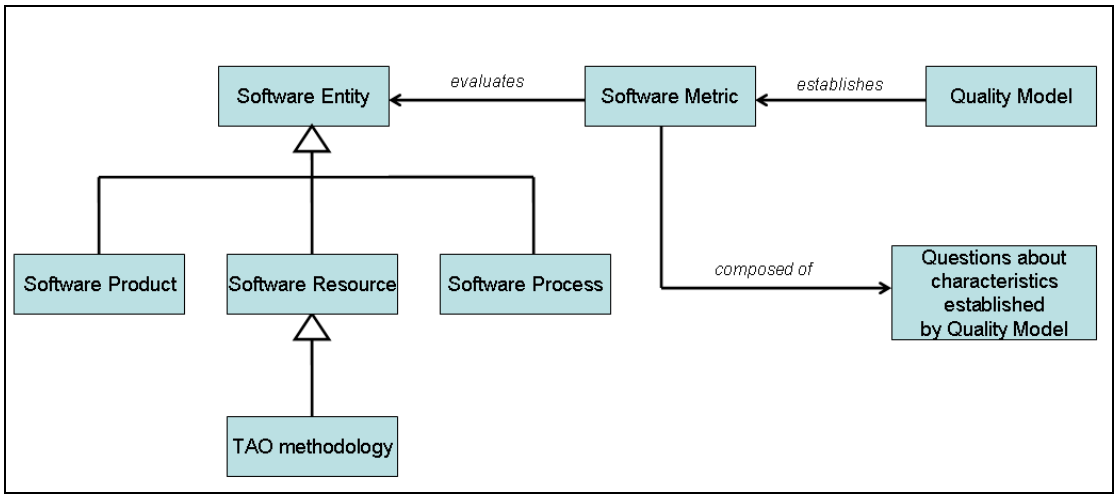

Figure 4-2 Software metrics composed of questions about characteristics established by quality models

The motivation of expanding the software metrics with a set of questions is that the answers to the established questions could measure if the software entity characteristics are fulfilled or not.

4.2.1 Usability metrics

The quality model ISO 9126 described before includes the usability as one of the characteristics that should be evaluated in any software entity. The usability characteristics are useful for measuring if user can save time and efforts when he uses the software entity. According to the usability sub-characteristics enumerated in section 3.2.1, hereafter a short explanation of each sub-characteristic is presented along with the set of questions for each sub-characteristic in order to establish the usability metrics addressed by TAO case studies for evaluating the TAO methodology, and hence, in order to be able to measure the fulfilment of TAO methodology for each usability metrics.

In order to enumerate and list the software entity sub-characteristics according to usability characteristics, and the questions for each software entity sub-characteristic, the following mnemonic code is used:

Mnemonic code	Meaning
UCx	Usability characteristic number *x*
QyUCx	Question number *y* about the usability characteristic number x

UC1. **Learnability**: the ability of the software entity to enable the user to learn it easily.

Q1UC1 What is the estimated and approximate time that users need to learn and use this methodology with respect to others methodologies?

Q2UC1 What is the number of tutorials and training courses necessary to make users learn the methodology?

UC2. **Understandability**: the ability of the software entity to enable the user to understand whether it is suitable, and how it should be used for particular tasks under certain conditions.

Q1UC2 In the ontology learning phase, is the TAO methodology providing guidelines to the ontology experts in order to help them to determine text-mining instances?

Q2UC2 Is the TAO methodology cookbook readable and understandable for engineers in the transitioning process?

Q3UC2 Is the ontology extraction performance and annotation performance understandable?

Q4UC2 In order to understand the TAO methodology, what is the necessary number of external teams that should help to use TAO methodology?

UC3. **Operability**: the ability of the software entity to enable the user to interact with it and control it easily.

Q1UC3 In the knowledge acquisition phase, is it easy (in terms of time) for domain engineers to define and collect all the relevant resources about the legacy application and load them into the TAO repository?

Q2UC3 How many resources are needed for transitioning a legacy system using the TAO methodology?

UC4. **Compliance/Conformance**: the ability of the software entity to be published and to be specified according to some standards, style guides, usability conventions, UML notations, etc. in order to improve the understandability of the software entity, from a user point of view (i.e. usability approach).

Q1UC4 Is the TAO methodology specified according to some standard notation?

Q2UC4 Is the resulted ontology consistent and compliant with standard ontological languages?

UC5. **Attraction**: the ability of the software entity to gain the users, taking into account for example the good results of some tasks or thanks to the time and effort saved when the user performs the tasks, etc.

Q1UC5 Have the ontologies built using the TAO methodology a better quality in contrast with the ontologies built without using the TAO methodology? Is the extracted ontology a good basis for further refinement?

Q2UC5 Is the ontology obtained after the ontology learning process a good result in a task-oriented view and is it helpful for the transitioning process?

Q3UC5 How many common transitioning and modelling mistakes can be avoided using the TAO methodology?

Q4UC5 Are the SWS descriptions a good basis for the service deployment?

Q5UC5 What it the estimated and approximate time needed for transitioning a legacy system using the TAO methodology?

Q6UC5 How many references to TAO methodology appear in papers or in other documents? What is the number of published papers about TAO methodology?

Q7UC5 Is it easy or complex/complicated to maintain the results obtained from the TAO transitioning process?

4.2.2 Technical metrics

The quality model ISO 9126 described before includes the functionality as one of the characteristics[2] that should be evaluated in any software entity. The functional characteristics are useful for measuring a set of software entity's functions that satisfy the specific and implied requirements in order to integrate the software entity with other software entities in terms of accuracy of discovery, level of support in integrating services within third party workflows, etc. According to the functional sub-characteristics enumerated in section 3.2.1, hereafter a short explanation of each sub-characteristic is presented along with the set of questions for each sub-characteristic in order to establish the usability metrics addressed by TAO case studies for evaluating the TAO methodology, and hence, in order to be able to measure the fulfilment of TAO methodology for each functional metric (also known as technical metric).

[2] Functionality characteristics are also known as technical characteristics

In order to enumerate and list the software entity sub-characteristics according to technical characteristics, and the questions for each software entity sub-characteristic, the following mnemonic code is used:

Mnemonic code	Meaning
TCx	Technical characteristic number *x*
QyTCx	Question number *y* about the technical characteristic number x

TC1. **Suitability**: the ability of the software entity to provide a set of appropriate functions for specific end-user tasks and objectives.

Q1TC1. Does the TAO methodology cover all the phases in the semi-automatic construction of SWS descriptions obtained from the transitioning process of legacy applications?

Q2TC1. In the knowledge acquisition phase, is the TAO methodology taking into account the data sources types that you need for your case study?

Q3TC1. Are the SWS descriptions covering the objective and functionalities desired in the transitioning process?

TC2. **Accuracy**: the ability of the software entity to provide rigorous and exact results and effects with a certain degree of precision.

Q1TC2. Is the whole legacy system domain covered by the ontology generated in the transitioning process?

Q2TC2. Do the extracted ontology and semantic annotated resources support a certain task, such as more effective query and answer?

Q3TC2. In the service and content augmentation phase, does the automated annotation provide an effective result? Are there a big number of missed or unnecessary annotations?

TC3. **Interoperability/Technological support**: ability of the software entity to interact with one or more specified systems and the ability of some tasks to be supported by tools (e.g. third party tools, etc).

Q1TC3. Regarding the degree of integration of WP2-WP5 results, what is the number of methods and tools integrated in the TAO methodology?

Q2TC3. What is the degree of integration with WP6 and WP7?

Q3TC3. How easy is it to extend the methodology with new activities?

Q4TC3. How easy is it to include new methods or techniques in an existing activity of the TAO methodology?

Q5TC3. What are the TAO methodology activities supported by TAO tools? Which are the TAO methodology activities which do not need to be necessarily supported by TAO tools?

Q6TC3. Are there different tools suggested by the TAO methodology for the different stages of the TAO transitioning process?

Q7TC3. Are these tools functionalities aligned with the objective of TAO methodology?

TC4. **Security**: the ability of the software entity to protect the data information of the application

Q1TC4. Is there some task that provides security to other tasks of TAO methodology?

Q2TC4. Do TAO tools provide some security related feature?

TC5. **Compliance/Conformance**: the ability of the software entity to be technologically adapted to standards, style guides, usability conventions, etc. from a functional point of view.

Q1TC5. Are the resulted ontologies formalized in a proper ontology language description approach and described using the style guide (e.g. naming convention)?

Q2TC5. Are the resulted SWS descriptions formalized in a proper service annotation approach?

In the following sections, the TAO case studies evaluate the TAO methodology based on the metrics proposed in section 4.2 by means of answering to the questions for each usability sub-characteristics and technical sub-characteristic.

4.3 Evaluation based on case studies

In this section firstly a brief description of each case studies is included, and afterwards, the set of usability and technical metrics established by the set of questions (see section 4.2), are answered by case studies partners along with the collaboration of ATOS. Consequently, the usability and technical characteristics will be measured by the level of fulfilment expressed in the answers to the questions.

4.3.1 Description of Case study 1: GATE

The aim of this case study is to provide a publicly available reference showcase of a re-factored ontology--based system, in a way that can facilitate its discovery and integration by other applications. GATE is a component-based framework for building and deploying Human Language Technology applications. The framework comprises a core library and a number of reusable Language Engineering plug-ins that are capable of performing basic language processing tasks such as POS tagging and semantic tagging. GATE uses plug-in mechanisms (based on JAR and XML configuration files) to access its resources, which are divided into three main categories namely Language Resources (LR), Processing Resources (PR), and Visual Resources (VR).

Language Resources are data-only resources such as documents, corpora, and ontologies. Processing Resources refers to programmatic/algorithmic entities such as parsers, generators and translators. Visual Resources are visualization and editing components that form the GATE GUI. Collectively these resources are known as a Collection of Reusable Objects and Language Engineering (CREOLE).

As resources for the first steps of the methodology, GATE provides a number of knowledge sources i.e. source-code, Javadoc, discussion forums, user manuals etc. that explicitly and implicitly provide a basic description of the underlying GATE functionalities and resources. Due to the diversity/volume of this information and lack of centralised indexing it is very difficult for new users to find answers to their questions or even start using the system. In addition, most of this knowledge sources are continuously updated due to new releases of software and functionality evolution. GATE also relies on media contents such as images, screen shots and UML diagrams

to describe the structure of the system and how it can be extended (i.e. customized) to specific needs.

As Web Services, currently GATE provides two types of services, each of which is associated with certain type of resources. These services are described in WSDL and can be accessed over HTTP. Further details of these can be found in D6.1 [8]

4.3.2 Description of Case study 2: DASSAV

DASSAV case study focuses on providing semantic-based support to aircraft maintenance processes. The goal of case study 2 is to show how maintenance legacy data repositories and business processes can be migrated into a semantic-powered SOA using TAO transitioning support and to practically assess the added value of the resulting service-based applications in terms of interoperability, knowledge reuse and adaptability. In the reengineered processes, ontology-based models of technical data and services will be exploited to ensure semantic interoperability between maintenance services and to provide standardised knowledge inputs for user mediated semantic annotation of technical publications.

The case study is composed of three inter-related scenarios:

- Semi-automated acquisition of the domain ontologies: constructing the fine-tuned ontologies needed to ensure interoperability is considered as a key step in a transitioning project. Significant efforts in the case study development are dedicated to acquisition of ontologies from legacy data, and more particularly from corporate relational databases.

- Content augmentation of technical publications: ontology-based annotation tools are exploited to support time-consuming content augmentation tasks performed manually in the legacy process.

- Content augmentation of maintenance services: An SAWSDL-based data mediation approach is applied to ensure semantic interoperability in a revisited scenario of aircraft maintenance management.

4.3.3 Usability metrics in the context of TAO case studies

This section includes the answers from both TAO case studies side about the question of usability characteristics explained in section 4.2.1, in order to evaluate the TAO methodology, and TAO transitioning process in general taking also into account the TAO tools.

In order to enumerate and list the software entity sub-characteristic according to usability characteristics, and the questions for each software entity sub-characteristic, the following mnemonic code is used:

Mnemonic code	Meaning
UCx	Usability characteristic number *x*
QyUCx	Question number *y* about the usability characteristic number x

AQyUCxCSz	Answer to question number *y* about the usability characteristic number x, in the context of TAO Case study *z* (if z=1 => TAO Case Study 1 GATE, if z=2 => TAO Case Study 2 DASSAV). Each answer has been provided by the TAO Case study *z* along with ATOS.

UC1. **Learnability**: the ability of the software entity to enable the user to learn it easily.

Q1UC1. What it the estimated and approximate time that users need to learn and use this methodology with respect to others methodologies?

AQ1UC1CS1. The GATE case study leader was proficient in Semantic Web technologies and also in web services before the start of the GATE case study. Therefore, this case study did not encounter significant problems in learning the TAO methodology on the basis of the deliverables provided. Although some of the tools used in different steps of the methodology were new and required some training, learning the methodology did not require any significant effort, as the deliverables from WP1 were good guidelines. In addition, there were also two discussion meetings and four or five email exchanges with the methodology partners.

AQ1UC1CS2. The DASSAV case study considered that best practices would be useful. In fact, in order to learn and make use of this methodology, it would be desirable more documentation about the TAO methodology, but obviously, it is reasonable that there is more documentation for other methodologies, because the TAO methodology is newer than the others. The TAO tutorial and the TAO methodology cookbook provide the documentation needed.

Q2UC1. What is the number of tutorials and training courses in order to users can learn the methodology?

AQ2UC1CS1. Ontology learning phase of the methodology required some familiarisation with the Ontology Learning tools provided by JSI (LATINO) and OntoGen. As the GATE case study leader was not previously exposed to neither of these tools, some training was required. For this purpose, online videos from the TAO WEB-site were used, in addition to four or five email messages exchanged with the WP2 leader (i.e. JSI). The deliverables from WP2 are used for the acknowledgement of more details. Another task from the methodology that required training was the web service annotation. However, due to the background knowledge of the GATE case study leader, and her understanding of the annotation process and the document structure (e.g. WSDL file), only basic guidelines were enough (received by email from the WP1 leader i.e. SOTON).

AQ2UC1CS2. DASSAV case study considered that it would be preferable to limit the number of tutorials needed to understand the methodology and to be ready to start applying it.

UC2. **Understandability**: the ability of the software entity to enable the user to understand whether it is suitable, and how it should be used for particular tasks under certain conditions.

Q1UC2. In the ontology learning phase, is the TAO methodology providing guidelines to the ontology experts in order to help them for determining text-mining instances?

AQ1UC2CS1. The TAO methodology details the design lifecycle of a domain ontology. The first two steps include Ontology Learning (OL) and Knowledge Acquisition (KA). OL is carried out using LATINO for data preparation and then ONTOGEN as a visual environment to interact with the user during the knowledge elicitation step. In the first iteration ONTOGEN gives suggestions for the top level of the ontology. And although a set of guidelines mentioned as examples are included in the page 53 of D1.2.v2 [2] about how to determine text-mining instances in the context of GATE case study, these could further be extended by more automatically derived suggestions for sub-concepts and instances. These suggestions are helpful for the domain expert who post-edits the ontology. An in-depth presentation of this process can be found in D6.2 [9].

AQ1UC2CS2. Most of the data in the DASSAV case study are coming from a legacy database, though some text-based sources like business standards are also relevant as complementary data sources. Consequently, a set of guidelines mentioned as examples are included in the page 53 of [2] about how determine the text-mining instances in the context of DASSAV case study.

Q2UC2. Is the TAO methodology cookbook readable and understandable for the engineers in the transitioning process?

AQ2UC2CS1. The knowledge engineers in the GATE case study did not encounter significant difficulties with the methodology documentation. In fact, in addition to the cookbook, another useful resource is the tutorial on the methodology, available at http://www.tao-project.eu/resources/eswc08-tao-tutorial.html

AQ2UC2CS2. The description of the main parts of the methodology through UML diagram is suitable.

Q3UC2. Is the ontology extraction performance and annotation performance understandable?

AQ3UC2CS1. Ontology extraction performance is evaluated by using this ontology in various scenarios. One of them is the knowledge coverage of the ontology which reveals the percentage of the questions posted on the GATE mailing list (randomly selected sample) can be actually answered using the developed ontology. Results presented in D6.4 (61.1%) are considered satisfying. Another evaluation is presented in D2.3. Annotation performance is described using standard Information Extraction measures, precision and recall. Results presented in both D6.4 and D3.3 are quite encouraging.

AQ3UC2CS2. It is important to make explicit the levels of performance that can be expected given the characteristics of the legacy application to be enhanced. For instance, regarding ontology learning, it can reasonably be

assumed that better accuracy can be expected from structured input like databases or XML data repositories than from raw text corpora.

Q4UC2. In order to understand the TAO methodology, what is the necessary number of external teams that should help to use TAO methodology?

AQ4UC2CS1. Tutorials available on the Web (links to http://www.tao-project.eu/resources/eswc08-tao-tutorial.html and also to videos from VideoLectures: http://videolectures.net/eswc08_wang_tla/) are enough for understanding and applying TAO methodology. Therefore, there is no need of any external team to help to use the TAO methodology. In addition, TAO showcase which is forthcoming as the part of WP8 and also the web-based navigator of TAO methodology cookbook (part of WP1) will be of a great importance for anyone interesting to use this methodology.

AQ4UC2CS2. It should be possible to internally conduct the transitioning process by following the TAO methodology, because DASSAV case study does not consist of an open source application, in contrast to GATE case study. However, the DASSAV case study needed a team which included at least an expert in semantic technologies.

UC3. **Operability**: the ability of the software entity to enable the user to interact with it and control it easily

Q1UC3. In the knowledge acquisition phase, is it easy (in term of time) for the domain engineers to define and collect all the relevant resources about the legacy application and load into the TAO repository?

AQ1UC3CS1. Software engineering case study like the GATE one is likely to have already collected in separate directories/URLs online the several main kinds of legacy content: source code, WSDL files, API documentation, manuals and email/forum/blog posts. Therefore, following the TAO methodology and applying the relevant techniques to different kinds of content is straightforward.

AQ1UC3CS2. For DASSAV case study, the support provided for knowledge acquisition is the most important asset that clearly shows the benefits of the TAO technology. And although TAO methodology considers any legacy resource to be collected, in the TAO methodology is explicitly considered the legacy resources managed by DASSAV Case Study, such as database, database scheme, WSDL files, and technical publications.

Q2UC3. How many resources are needed for transitioning a legacy system using the TAO methodology?

AQ2UC3CS1. Basically, the main resources for transitioning GATE using the TAO methodology, are: the domain expert from GATE, the legacy data from GATE and the TAO Suite which integrates the TAO components (along with the appropriate LATINO adapter for using it with TAO Suite) and consequently implements the TAO methodology

AQ2UC3CS2. Although the software support significantly contributes to reduce the effort required from experts, validation by maintenance experts remains a critical step of the transitioning process. Consequently, in addition

to the domain expert from DASSAV, and the legacy data from DASASV case study, some software tool, such as TAO Suite, is needed in order to reduce the effort of transitioning the DASSAV legacy system.

UC4. **Compliance/Conformance**: the ability of the software entity to be published and to be specified according to some standards, style guides, usability conventions, UML notations, etc. in order to improve the understandability of software entity, from a user point of view (i.e. usability approach).

Q1UC4. Is the TAO methodology specified according to some standard notation?

AQ1UC4CS1. The TAO methodology cookbook is graphically depicted by UML in a general detail by describing activities, flow conditions, etc. However, it would be desirable that the notation of TAO methodology cookbook will be more UML-compliant according to activities' fork, etc. For this, it is necessary to see the UML 2.0 standard. However, the current version of UML-style used in the TAO methodology cookbook is enough in order to be used by GATE case study.

AQ1UC4CS2. Since the TAO methodology is intended to provide high-level guidelines, there is no need to make it strictly compliant with modelling standards like UML. Combination of textual description with UML-style diagrams allows reaching an acceptable level of formalisation.

Q2UC4. Is the resulted ontology consistent and compliant with standard ontological languages?

AQ2UC4CS1. Since TAO tools which are integrated by TAO Suite are OWL-compliant, the ontology generated will be consistent and compliant with OWL language.

AQ2UC4CS2. One of the advantages of semantic technologies is the ability to incorporate in the process in-built consistency checking facilities based on reasoning capabilities. Thanks to generation of OWL ontologies by means of integration of LATINO into TAO Suite, we can expect that the resulted ontology will be a consistent conceptual model.

UC5. **Attraction**: the ability of the software entity to gain the users, taking into account for example the good results of some tasks or thanks to the time and effort saved when the user performs the tasks, etc.

Q1UC5. Have the ontologies built using the TAO methodology a better quality in contrast to the ontologies built without using the TAO methodology? Is the extracted ontology a good basis for further refinement?

AQ1UC5CS1. The main advantage of the TAO ontology extraction phase is that it is generally automatic, which lowers the cost of bootstrapping the domain ontology and also makes it easier not to miss important domain concepts. The methodology also recommends refining manually the quality of the extracted ontology further by using an ontology editor of choice. Automatically derived ontology is not perfect, and serves more as a guideline for the domain expert whose role is to refine it and create a gold standard. Therefore, it is not feasible to use ontology learning tools in very narrow domains, where it would be easier to create ontology from scratch.

AQ1UC5CS2. The TAO methodology takes into account various scenarios involving more or less automated derivation of the ontologies from legacy data, which is a good advantage. For instance, reusing of existing ontologies is considered in the early steps of the process supported by the methodology. This is an important aspect since reusability of the produced ontologies is a key quality dimension. Consequently, thanks to these features, it is clear that the ontologies built semi-automatically using the TAO methodology have at least the same or better quality in contrast to the ontology built without using the TAO methodology.

Q2UC5. Is the ontology obtained after the ontology learning process a good result in a task-oriented view and is it helpful for the transitioning process?

AQ2UC5CS1. After manually refining the automatic generated ontology, this ontology will have a better quality, and consequently it will be a good and helpful result in the transitioning process. An exact quantification of the improvements and performance of ontology learning is provided in the forthcoming D2.3 “Ontology learning: Evaluation Report”.

AQ2UC5CS2. Performance of ontology learning can be very high on some data sources, as in DASSAV case study. In some context where further significant manual refinement is needed, it is at least a good way to avoid the blank page syndrome when transitioning a legacy application.

Q3UC5. How many common transitioning and modelling mistakes can be avoided using the TAO methodology?

AQ3UC5CS1. Although methodology covers both fully automatic and semi-automatic ontology generation, in cases when the domain expert uses the automatic transitioning process only, several mistakes in the generation of semantic artefacts might occur. Hence, semi-automated approach is desirable so that the domain expert can use external tools in order to manually refine the generated semantic artefacts.

AQ3UC5CS2. Since the transitioning help is heuristic-based in essence, it is important to stress that the automatically generated models should be carefully verified and refined by semantic experts. Hence, consistency check helps to identify odd results.

Q4UC5. Are the SWS descriptions a good basis for the service deploying?

AQ4UC5CS1. The TAO Suite and TAO methodology decrease the cost for users undertaking semantic annotation of web services by automating part of the process. However, as service deployment requires 100% correctness and completeness of the semantic service descriptions, a manual refinement step is necessary. This can be achieved either with the help of the annotation editing tools from WP3 (see D3.4 “User tools”) or with third-party tools, produced outside of the TAO project, e.g., Radiant.

AQ4UC5CS2. In DASSAV case study, semantic descriptions of web services are only used to ensure a high interoperability level in service exchanges (through SAWSDL data mediation). However, it is recommendable to refine the generated SWS descriptions by external tools (e.g. Radiant) in order to ensure the correctness of automatically generated SWS descriptions.

Q5UC5. What it the estimated and approximate time needed for transitioning a legacy system using the TAO methodology?

AQ5UC5CS1. The time required depends primarily on whether the application requires 100% correctness of the semantically augmented legacy content. In cases such as the GATE case study, it is viable to gain significant added value from the technology without significant manual post-editing (except for refinement of the learnt domain ontology and the generated SA-WSDL). Computing time required by the ontology learning and content augmentation may also become a factor, if there is a significant amount of data to process. The GATE case study experimented with source code, WSDL files, manuals, and user forum postings and these could be processed without a major computational overhead. The case study runs a re-indexing process each night, which has been successful so far, but, if the need arises in the future, it is possible to parallelize the content augmentation tools to process different types of legacy data on different computers.

AQ5UC5CS2. In DASSAV case study, migration of data models and data is made easier by the tools. However, migration of the processes involves much more manual effort.

Q8UC5 How many references to TAO methodology appear in papers or in other documents? What is the number of published papers about TAO methodology?

Answered jointly by both TAO case studies and SOTON. The applied transitioning process is described in two papers: one published in the proceedings of ESTC 2007 conference on semantic technologies and the other one in IBIS-journal (Interoperability in Business Information Systems). Additionally, several of TAO deliverables refer to TAO methodology cookbook (e.g. this current deliverable), and the TAO partners are encouraged to cite the TAO methodology in their publications. In fact, the partners from an external project, ServiceFinder, are interested in the work of TAO, and specially, in the work of TAO methodology, although they did not make any citations in their current version of deliverable yet.

Q6UC5. Is it easy or complicate to maintain the results obtained from the TAO transitioning process?

AQ6UC5CS1. Due to the fact that TAO Suite provides functionalities about importing and exporting the resulted semantic artefacts in the transitioning process, it is expected that the resulted semantic artefacts will be relatively easy to be maintained with the aforementioned import and export functionalities in order to refine these results with external tools. In fact, in order to ensure maintainability, the TAO methodology and TAO Suite are based on established or newly emerging standards, such as OWL and SA-WSDL.

AQ6UC5CS2. Management facilities offered in the new framework (i.e. TAO Suite) allow implementing some maintenance functionalities by means of exporting generated results and importing the results refined by external tools.

4.3.4 Technical metrics in the context of TAO case studies

This section includes the answers from both TAO case studies side about the question of functional characteristics explained in section 4.2.2, in order to evaluate the TAO methodology, and TAO transitioning process in general taking also into account the TAO tools.

In order to enumerate and list the software entity sub-characteristics according to usability characteristics, and the questions for each software entity sub-characteristic, the following mnemonic code is used:

Mnemonic code	Meaning
TCx	Technical characteristic number *x*
QyTCx	Question number *y* about the technical characteristic number x
AQyTCxCSz	Answer to question number *y* about the technical characteristic number x, in the context of TAO Case study *z* (if z=1 => TAO Case Study 1 GATE, if z=2 => TAO Case Study 2 DASSAV). Each answer has been provided by the TAO Case study *z* along with ATOS.

TC1. **Suitability**: the ability of the software entity to provide a set of appropriate functions for specific end-user tasks and objectives.

Q1TC1. Does the TAO methodology cover all the phases in the semi-automatic construction of SWS descriptions obtained from the transitioning process of legacy applications?

AQ1TC1CS1. The three main tasks necessary for GATE case study are knowledge acquisition by storing and classifying legacy data (e.g., source code, manuals), ontology learning from that legacy data, and semantic annotation of legacy data and services. All these three phases are well supported by the TAO methodology and TAO Suite.

AQ1TC1CS2. The TAO transitioning process provides the needed high-level guidelines to support the various steps of DASSAV cases study transitioning work.

Q2TC1. In the knowledge acquisition phase, is the TAO methodology taking into account the data sources types that you need for your case study?

AQ2TC1CS1. Yes, the TAO methodology is generic as it allows any legacy data source to be used. The LATINO tool explicitly considers ontology learning from the legacy resources managed by the GATE Case Study, such as Java Source Code, API documentation, and user's manual.

AQ2TC1CS2. The main types of data sources involved in the DASSAV case study are considered in the TAO methodology.

Q3TC1. Are the SWS descriptions covering the objective and functionalities desired in the transitioning process?

AQ3TC1CS1. Yes, the SWS descriptions created in the transitioning process are useful for the GATE case study in order to discover the hidden knowledge of GATE services and functionalities. However, the scope of the GATE case study is such that it does not plan to execute the generated SWS.

AQ3TC1CS2. With respect to DASSAV case study requirements, SWS descriptions created in TAO transitioning process are useful to solve data heterogeneity problems between services involved in maintenance processes of DASSAV case study.

TC2. **Accuracy**: the ability of the software entity to provide rigorous and exact results and effects with a certain degree of precision

Q1TC2. Is the whole legacy system domain covered by the ontology generated in the transitioning process?

AQ1TC2CS1. Ontology covers the most important GATE components including Visual Resources, Processing Resources and Language Resources. Evaluation with users (details in D6.4) reveals that in more than 50% of the time, the users could find the relevant answer to the question (related to finding the specific GATE component) using the showcase based on the ontology developed through the transitioning process.

AQ1TC2CS2. The TAO methodology covers most parts of the process. The process is refined for dealing with the specific problems of transitioning relational databases and reuse of business standards.

Q2TC2. Does the extracted ontology and semantic annotated resources support a certain task, such as more effective query and answer?

AQ2TC2CS1. As detailed in D6.4, the extracted ontology was of a good quality as it covers more than 60% of knowledge about GATE. This is concluded from the previously mentioned ontology coverage evaluation, and also from the user-centric evaluation where the users reported if the results of searching for the specific GATE components were relevant or not.

AQ2TC2CS2. It was difficult to evaluate this feature, but in most of the cases the query is effective.

Q3TC2. In the service and content augmentation phase, does the automated annotation provide an effective result? Are there a big number of missed or unnecessary annotations?

AQ3TC2CS1. Automatic annotation provided reasonable results as detailed in D3.3. Overall precision and recall figures were in the range from 80-100%, which is above what we have expected.

AQ3TC2CS2. Automated annotation is a convenient functionality. However, results should be carefully controlled through a dedicated user interface.

TC3. **Interoperability/Technological support**: ability of the software entity to interact with one or more specified systems and the ability of some task to be supported by the tools (e.g. third party tools, etc).

Q1TC3. Regarding the degree of integration of WP2-WP5 results, what is the number of methods and tools integrated in the TAO methodology?

AQ1TC3CS1. All tools except the ones such as the export and import functionalities, and the usage of external tools in order to refine the generated semantic artefacts in the transitioning process.

AQ1TC3CS2. From the perspective of DASSAV case study, the methodology generally provides instructions for utilising the tools of the TAO suite.

Q2TC3. What is the degree of integration with WP6 and WP7?

AQ2TC3CS1. Both case studies have been following the TAO methodology steps, mainly based on the information included in relevant deliverables from WP1, and also discussions and instructions from WP1 partners. Overall, the current focus of the TAO methodology is well addressed.

AQ2TC3CS2. The relevant entry points in the methodology are well identified for transitioning the application of DASSAV case study.

Q3TC3. How easy is it to extend the methodology with new activities?

AQ3TC3CS1. Since the TAO methodology is contained in a cookbook and graphically represented with UML notation, the set of sequential activities could be easily extended in order to specify more activities and sub-activities.

AQ3TC3CS2. As requested, the current version of the methodology includes references to transition relational databases to ontologies, which is very important for the DASSAV case study.

Q4TC3. How easy is it to include new methods or techniques in an existing activity of the TAO methodology?

AQ4TC3CS1. As already mentioned in the previous question, thanks to the fact that TAO methodology is contained in a cookbook and graphically represented in UML, and taking into account the sub-activities and methods of each activity, the set of methods and techniques could be easily extended.

AQ4TC3CS2. The methodology seems to be in a form that makes it easily extensible.

Q5TC3. What are the TAO methodology activities supported by TAO tools? What are the TAO methodology activities which do not need to be necessarily supported by TAO tools?

Answered by both case studies. The knowledge acquisition activity and usage of the TAO repository involve the HKS (WP4) and TAO Suite (WP5). The ontology learning activity involves LATINO (WP2) and TAO Suite. And the service and content augmentation activity involves the CA MANAGER (mainly belonged to WP5, but also to WP3) and TAO Suite. Consequently, all TAO methodology activities are supported by TAO tools, although TAO methodology defines that the content augmentation of documents is performed in parallel along with the content augmentation of web services, and the TAO tools sequentially performs these two steps due to usability reasons in order to making easier the interactions for the user.

Q6TC3. Are there different tools suggested by the TAO methodology for the different stages of the TAO transitioning process?

AQ6TC3CS1. TAO methodology is not primarly concerned with the tools which could be used to support all transitioning steps, but more with guidelines to what can be used to transition legacy data to ontologies and semantic services. TAO Suite supports each transitioning step..although the methodology could be used as a guideline for the transitioning using some external tools if available.

AQ6TC3CS2. The TAO methodology is built up on the envisaged functionalities of the tools developed in TAO work packages.

Q7TC3. Are these tool functionalities aligned with the objective of TAO methodology?

AQ7TC3CS1. Correct at least for the knowledge acquisition and ontology learning activities. However, the TAO methodology considers that semantic annotation of documents is performed in parallel with respect to semantic annotation of services. However, the TAO Suite performs semantic annotation of documents in a sequential way with respect to the functionality about semantic annotation of services, so that, firstly the semantic annotation of documents is performed, and afterwards the creation of semantic web services is performed.

AQ7TC3CS2. TAO tools and TAO methodology stages seem to be aligned in a consistent way.

TC4. **Security**: the ability of the software entity to protect the data information of the application.

Q1TC4. Is there some task which provides security to the other tasks of TAO methodology?

AQ1TC4CS1. The TAO methodology is not concerned with security at present, but it could be included easily by means of including in the initial point of TAO methodology a sub task about creating a new project for each transitioning process, as supported by the TAO Suite (user authentication & project registration by setting some password before the knowledge acquisition).

AQ1TC4CS2. Security aspects are not addressed by TAO methodology. But it could be a good decision that they will be addressed in future versions or in future projects.

Q2TC4. Do TAO tools provide some security related feature?

Answered by both TAO case studies. The TAO Suite provides a first step, before knowledge acquisition in order to perform user authentication and project registration by setting user name and password in addition to other complementary data. Where possible, the TAO components will provide some security for invoking their functionalities with this authentication data.

TC5. **Compliance/Conformance**: the ability of the software entity to be technologically adapted to standards, style guides, usability conventions, etc. from a functional point of view.

Q1TC5. Are the resulted ontologies formalized in a proper ontology language description approach and described using the style guide (e.g. naming convention)?

Answered by both TAO case studies. The resulting ontologies are OWL compliant, and they can be refined with any OWL compliant tool.

Q2TC5. Are the resulted SWS descriptions formalized in a proper service annotation approach?

Answered by both TAO case studies. The resulting SWS descriptions are SA-WSDL compliant and they can be refined with any SA-WSDL compliant tool

5 Analysis of the results and conclusion

5.1 Analysis of the results

The evaluation of the final TAO methodology has been performed by collecting the answers provided by TAO case studies leaders. In order to check the progress of the methodology, we kept the same set of questions envisaged in first version of this deliverable (D1.3.2v1). The analysis of the answers to these questions shows the level of fulfilment of the usability and technical metrics

Usability metrics analysis:

Learnability: In the first version of this deliverable, users of the GATE case study were already satisfied with the documentation and easyness of use of the TAO tools. However, users from the DASSAV case study claimed more sophisticated tutorials and the delivery of the methodology cookbook. This documentation has been produced during the last months of the project. Therefore the users perceived an improvement on this particular aspect.

Understandability: This metric is measuring the ability of the TAO tools to enable the user to understand whether it is suitable, and how it should be used for particular tasks under certain conditions. As in the previous metric, the delivery of extra training material has been perceived positively by the users in order to reach a better understandability of the system. The GATE users find that the ontology learning methodology is quite useful and they documented it in several deliverables. The DASSAV users, more focused in transitioning structured data from databases or XML to ontologies, find that the guidelines included in the methodology are suitable for this particular transitioning process. On the other hand, they would like to see clearer accuracy indicators for this type of transitioning process. Both case study users find clear and suitable the methodology cookbook and other methodology documentation.

Operability: The GATE users find easy to control the knowledge acquisition phase, because being a software engineering case study they had all their resources previously identified. In the DASSAV user opinion, the support provided for knowledge acquisition is the most important asset that clearly shows the benefits of the TAO technology. They also believe that the validation by maintenance experts remains a critical step of the transitioning process. Both case studies needed at least a domain expert.

Compliance/Conformance: The TAO Methodology is graphically depicted by UML in a general detail by describing activities, flow conditions, etc. GATE users would have liked a more UML 2.0 approach, although they find the current version easily understandable. DASSAV users did not require such a level of compliance with UML, finding that a good balance between UML notation and textual description is a good compromise. Regarding ontologies, both case studies are happy that the ontologies created by the TAO tools are OWL-compliant.

Attraction: Users from both case studies think that the automation of the extraction phase is the most attractive feature of TAO. However, this extraction cannot be fully automated, because automatically derived ontologies are not perfect. GATE users believe that in the case of very narrow domains, it would be easier to create ontologies

from scratch rather than using ontology learning tools. In the opinion of both case studies, the quality of the ontologies, once they are revised after the automatic creation process, is better than in a fully manual process. This issue has been particularly stressed by DASSAV users, because in their case they started using structured resources, which lead to an easier extraction process. Regarding SWS, both case studies agree on the need of using 3rd party tools (i.e. Radiant) in order to refine and generate the final SWS descriptions. They see the results of TAO as a good first automated step towards the SWS creation.

Technical metrics analysis:

Suitability: Both case studies agree on the perception that the TAO methodology covers their needs regarding the semiautomatic construction of SWS descriptions.

Accuracy: In both case studies, the accuracy of the results is considered good. In the case of GATE, they said that in more than 50% of the cases users are able to find relevant answers when using the learned ontology, while the ontology covers around 60% of the available GATE knowledge resources. GATE users also calculated from 80 to 100% of accuracy in the content augmentation phase, which is a higher precision than we expected.

Interoperability/Technological support: Both case studies agreed on the impression that the TAO Suite achieved a good degree of integration of the TAO results. On the other hand, the late integration of some of the tools did not give margin to the end users to test the final version. The delivery of the methodology cookbook has been positively appreciated by the users, as it shows the integration and the interoperability between the different TAO tools.

Security: Security is not a primary concern in TAO. Therefore, users find that although desirable in some aspects, it is not a must.

Compliance/Conformance: The use of OWL-compliant ontologies and the SA-WSDL-compliant SWS descriptions has been positively valued by the users.

5.2 Conclusion

Users from the case studies stated that the TAO methodology defines the integration and the usage path of the methods and tools developed by the different technical work packages. Every activity defined in TAO methodology is supported and aligned with the TAO methods and tools. In general terms, both case studies believe that the TAO methodology is useful for them.

Regarding time and resources needed for transitioning a legacy product with the TAO methodology, although the methodology provides guidelines to ease the process, it is clear that in some cases it will depend on the experience of the user, and on the amount of legacy data contained in the legacy product. Nevertheless the TAO methodology is clearly an important asset for the users in order to perform the transitioning process.

The delivery of the **TAO cookbook and tutorials on the web** has been highlighted by the users as one of the major outcomes of the last version of the methodology. Tutorials can be found at:

- http://www.tao-project.eu/resources/eswc08-tao-tutorial.html
- http://videolectures.net/eswc08_wang_tla/)

We may conclude that in general the evaluation results can be considered positive. According to the answers, the feedback from the first version of this deliverable seems to have been taken into account in the latest version of the TAO Methodology. Some of the weak points spotted previously have been solved, especially the ones concerning the need for more documentation and on-line training material,

6 Appendixes

6.1 Terminology

This appendix explains the terms followed in this whole report in order to solve any ambiguity caused by possible double meaning of some terms or by some other misunderstanding.

- TAO. It refers to the project called Transitioning Applications to Ontologies.
- TAO methodology. It corresponds with the SWS bootstrapping methodology defined in work package 1.
- Work package. It refers to a work package of TAO.
- M12. It means the month number 12 since TAO begins. Consequently, we can deduce the meaning of M24, M36 and so on.
- D1.3.v1. It refers to first version of deliverable report identified as 1.3, according to deliverable list of DoW. Consequently, we can deduce the meaning of D1.2, D5.2, and so on.
- TAO components. It refers to the following three components developed in the scope of TAO: LATINO, CA MANAGER and HKS.
- TAO Suite. It is the software infrastructure which shall integrate the three TAO components for supporting the semi automatic transitioning process of legacy systems into semantic-based systems.
- Transitioning process. Consists of migration steps for generating semantic-based systems from legacy systems.
- Case study. It refers to the applicability of the project results to some practical case. In TAO, there are two case studies and they are represented by work package 6 (case study 1: Transitioning a Large-Scale Open-Source System) and work package 7 (case study 2: Transitioning Business Process Applications). The case study 1 describes the transitioning of the GATE application provided by USFD partner. Case study 2 describes the transitioning of the application provided by DASSAV partner. Hence, case study 1 is also known as GATE case study, and case study 2 is also known as DASSAV case study. For TAO evaluation purposes, both the application from GATE case study and the application from DASSAV case study are considered as legacy systems.

Bibliography and references

This part contains the list of documents and other key references relevant to the deliverable.

[1] Paul Goodman, 1993, "*Practical Implementation of Software Metrics*", McGraw Hill, London

[2] Amardeilh, Florence; Vatant, Bernard; Gibbins, Nicholas; Payne, Terry R.; Saleh, Ahmed; H.Wang, Hai. "*SWS Bootstrapping Methodology*". Deliverable D1.2.v2 of TAO Project (IST-2004-026460).

[3] Amardeilh, Florence; Vatant, Bernard; Gibbins, Nicholas; Payne, Terry R.; Saleh, Ahmed; H.Wang, Hai. "*SWS Bootstrapping Methodology*". Deliverable D1.2.v3 of TAO Project (IST-2004-026460).

[4] Amardeilh, Florence; Vatant, Bernard; Gibbins, Nicholas; Payne, Terry R.; Saleh, Ahmed; H.Wang, Hai. "*SWS Bootstrapping Methodology*". Deliverable D1.2.v1 of TAO Project (IST-2004-026460).

[5] Martin, J.; Herrero, G.; Bontcheva, K.; Damljanovic, D.; Cerbah, F.;. "*Case study reports on evaluating the methodology*". Deliverable D1.3.v1 of TAO Project (IST-2004-026460).

[6] Martín, Jesús; Herrero, Germán; Capellini, Alberto; Francart, Thomas; Amardeilh, Florence; Marinova, Zlatina. "*Architecture and integration requirements and specifications*". Deliverable D5.2 of TAO Project (IST-2004-026460).

[7] ISO: "*International Organization for Standardization*". http://www.iso.org/

[8] Bontcheva, Kalina; Roberts, Ian; Agatonovic, Milan; Nioche, Julien; Sun, James. "*Case Study 1: Requirement analysis and application of TAO methodology in data intensive applications*". Deliverable D6.1 of TAO Project (IST-2004-026460).

[9] Damljanovic, Danica ; Bontcheva, Kalina ; Tablan, Valentin; Roberts, Ian; Agatonovic, Milan ; Shafirin, Andrey; Sun, James. "*Case study 1: Domain ontology and semantic augmentation of legacy content*". Deliverable D6.2 of TAO Project (IST-2004-026460).

[10] Niessink, F. (2002), "*Software requirements: functional and non-functional software requirements*", available at: www.cs.uu.nl/docs/vakken/swa/Slides/SA-2-Requirements.pdf

[11] Carvallo, J.P.; Franch, X.; Quer, C. "*Defining a Quality Model for Mail Servers*". Second International Conference on COTS-Based Software Systems, ICCBSS 2003, Ottawa, Canada.

[12] Botella, P.; Burgués, X.; Carvallo, J.P.; Franch, X.; Pastor, J.A.; Quer, C. "*Towards a Quality Model for the Selection of ERP Systems*". Component-Based Software Quality. Methods and Techniques. LNCS 2693. Eds: Cechich, A.; Piattini, M.; Vallecillo, A. Springer-Verlag, 2003. Page(s): 225- 246.

[13] Cai, X., Lyu, M.R., Wong, K-F. and Ko, R. (2000). Component-Based Software Engineering: Technologies, Development Frameworks, and Quality Assurance Schemes. Proc. of the Seventh Asia-Pacific Software Engineering Conference (APSEC'00), IEEE Computer Society, 372-379

[14] Rossi, P. Fernandez, G. "*Definition and validation of design metrics for distributed applications*". Sch. of Comput. Sci. & Inf. Technol., RMIT

Univ., Melbourne, Vic., Australia; This paper appears in: Software Metrics Symposium, 2003. Proceedings. Ninth International Publication Date: 3-5 Sept. 2003 On page(s): 124- 132 ISSN: 1530-1435 ISBN: 0-7695-1987-3 INSPEC Accession Number: 7854695

[15] Hangjung Zo, and Ramamurhty K. (2002) "*A Choice Model For Asssessing and Selecting E-Commerce Websites in a B2C Environment*". Eighth Americas Conference on Information Systems, pp. 348-355

[16] Parasuraman A., Zeithami V.A., and Berry L.L. (1998) "*SERVQUAL: a multi-item scale for measuring consumer perceptions of service quality*". Journal of Retailing 67(4), pp. 420-450

[17] Webb H.W., and Webb L.A. (2002). "*B2C Electronic Commerce Websites: an Analysis of Quality Factors*". Eighth Americas Conference on Information Systems, pp. 340-347.

[18] M. F. Bertoa and A. Vallecillo, "*Quality attributes for cots components*," in Proceedings of the 6th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2002).