D12.1 Requirements for GF based Verbalization in Be Informed

Contract no.	FP7-ICT-247914
Project full title:	MOLTO - Multilingual Online Translation
Deliverable:	D12.1 Requirements for GF based Verbalization in Be Informed
Security (distribution level):	Public
Contractual date of delivery:	1 September 2012
Actual date of delivery:	14 September 2012
Type:	Report
Status & Version	Final, 1.0
Author(s)	Joris van Aart, Jeroen Daanen, Jeroen van Grondelle, Menno Gulpers, Emiel van Haandel, Herko ter Horst, Xander Uiterlinden
Task Responsible	Be Informed

Abstract

This document outlines the requirements that need to be addressed when developing a verbalization component for Be Informed based on MOLTO Technologies in WP12 of the MOLTO project.

It outlines a number of scenario's where verbalization will be used and bases requirements on these. In addition, requirements are presented in areas such as Human Computer Interaction (HCI) and non-functional and technical requirements.

Table of Contents

1 Introduction

1.1 Background

As the adoption of ontologies into enterprise application environments grows, new audiences have to deal with ontologies, other than knowledge engineers and ontologists. These audiences range from business users, who need to take ownership of the ontologies, to end users, such as customers or citizens, who are presented with the services based on these ontologies. As the formalisms themselves are often inaccessible to these new audiences, appropriate visualizations are important. Our experience in practice is that business users often overcome their perception of graph-oriented visualizations being too technical when gaining experience. However, graph visualizations remain a challenge for incidental reviewers and end users. Therefore, verbalization of ontologies into natural language is one of the approaches that is crucial to make ontologies accessible to new audiences.

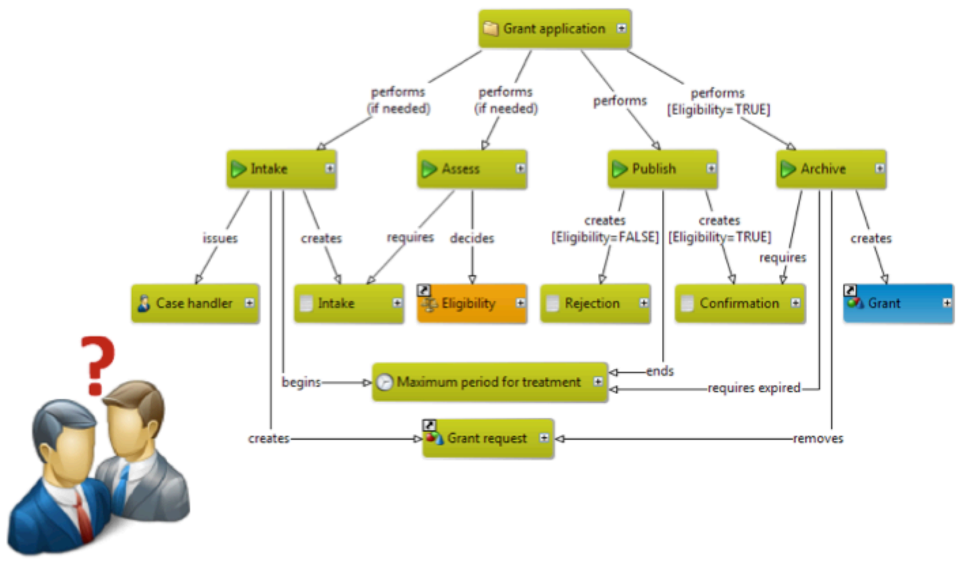


Figure 1. Poor Business User Adoption of Graphical Visualisations

Additionally, being able to provide verbalization in a multilingual manner is important: Governments and enterprise often offer their products and services in international contexts or to customers of different languages. For instance, Dutch Immigrations offers many of its services based on ontologies [ESWC2009], and it typically needs to interface with people that do not speak Dutch. Also, governments have to deal with numerous international aspects in legislation when drafting their national laws. Specifically in Europe, large parts of national legislation are either heavily influenced by or originates in European legislation. Being able to share ontologies capturing such international legislation and being able to refer to them from local ontologies offers important benefits in areas of productivity and traceability across local practices.

In 2010 Be Informed has developed a verbalization component based on pattern sentences, that is released as part of our product. It is discussed in detail in [EKAW2010] and [CNL2010].

The areas that need improving outlined in specifically [CNL2010] triggered our participation in the MOLTO Project.

1.2 About this Document

This document outlines the requirements that we will need to address when developing a verbalization component for Be Informed based on MOLTO Technologies in WP12 of the MOLTO project.

We have chosen a broad, slightly informal style of requirement capturing. We believe it improves readability and will make the document relevant for broader audiences. We have tried to capture requirements from a large number of perspectives. Some requirements apply to the verbalization component to be developed in WP12, but many also apply to the functionality that can be based on this component. Although out of scope for WP12, we believe it is the best way to visualize intended use and capture the inherently implicit requirements that this might pose on a technology we do not completely master at this time.

No formal distinction between must have and optional requirements is made. We believe the document will guide us in leveraging GF to the maximal extend in the development of a verbalization component in WP12.

We will use it for WP12 planning and resourcing, both within Be Informed and in discussions with Chalmers University concerning its role in WP12.

We will also use it when designing the verbalization component based on GF and the grammars for our four default modeling domains.

This document does not contain a detailed design of the grammars or the verbalization component, but rather the requirements the grammars should address.

1.3 Contributors

Editor of this document is Jeroen van Grondelle.

Contributing authors are Joris van Aart, Jeroen Daanen, Jeroen van Grondelle, Menno Gulpers, Emiel van Haandel, Herko ter Horst and Xander Uiterlinden.

Please direct questions, contributions and ideas to Jeroen van Grondelle at j.vangrondelle@beinformed.com.

2 Usage Scenarios

Be Informed captures policy in ontologies. These ontologies are used throughout the policy lifecycle from choosing/deciding on policy, communicating the agreed upon policy to all stakeholders to running the supporting applications.

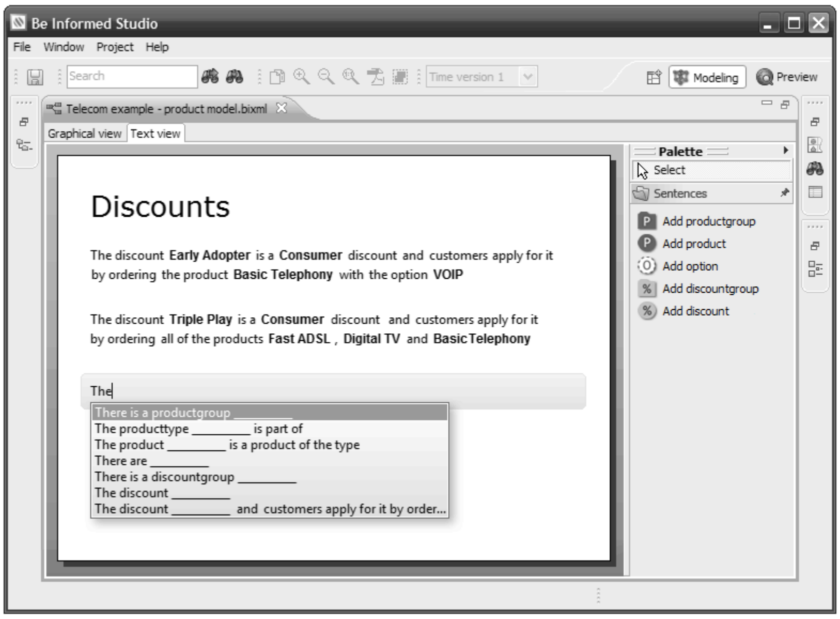
As a consequence, verbalizations of these ontologies could be used in a number of scenarios throughout that policy lifecycle. These scenarios have been used in requirement analysis to capture the problem domain that needs to be supported and captured in the requirements.

2.1 Review, Validation and Feedback of Models

For the ontologies to be used as the basis of actual applications, it is crucial they contain a correct representation of the requirements and constraints. Review and validation before deploying and the ability to provide feedback on the model after deployment is very important. A natural language representation of the models can help stakeholders to exercise these tasks. Special verbalization choices might have to be made to create texts that are effective in this specific scenario.

2.2 Text based Editing of Models

The most effective way of business user involvement is of course allowing them to create models themselves or, often more realistic, to maintain and alter existing models. In [EKAW2010] we explored editors that do use a textual metaphor to present models to the users, but that do not use typing text as editing metaphor.



Discounts

The discount **Early Adopter** is a **Consumer** discount and customers apply for it by ordering the product **Basic Telephony** with the option **VOIP**

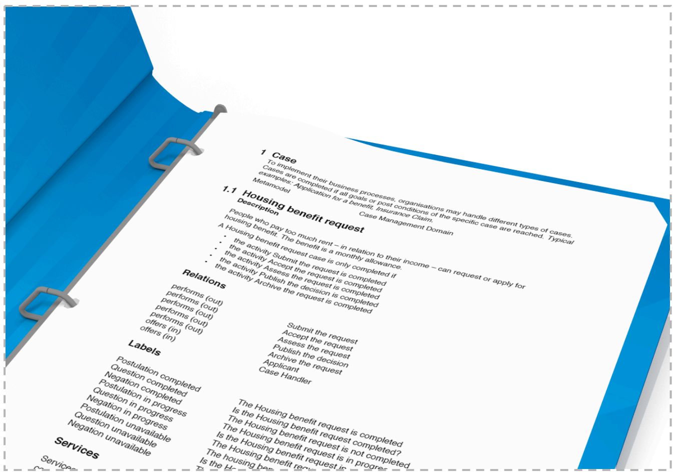
The discount **Triple Play** is a **Consumer** discount and customers apply for it by ordering all of the products **Fast ADSL** , **Digital TV** and **Basic Telephony**

The

There is a productgroup ________
The producttype ________ is part of
The product ________ is a product of the type
There are ________
There is a discountgroup ________
The discount ________
The discount ________ and customers apply for it by order...

2.3 Self Documenting Models

Typically, systems need to be well documented for IT organizations to be able to support production use and perform maintenance. The online, navigational access to the models is then often not acceptable, and conventional documentation sets need to be generated.



1 Case

To implement their business processes, organisations may handle different types of cases. Cases are completed if all goals or post conditions of the specific case are reached. *Typical examples: Application for a benefit, Insurance Claim.*

Metamodel Case Management Domain

1.1 Housing benefit request

Description

People who pay too much rent – in relation to their income – can request or apply for housing benefit. The benefit is a monthly allowance.

A Housing benefit request case is only completed if

- the activity Submit the request is completed
- the activity Accept the request is completed
- the activity Assess the request is completed
- the activity Publish the decision is completed
- the activity Archive the request is completed

Relations

performs (out)	Submit the request
performs (out)	Accept the request
performs (out)	Assess the request
performs (out)	Publish the decision
performs (out)	Archive the request
offers (in)	Applicant
offers (in)	Case Handler

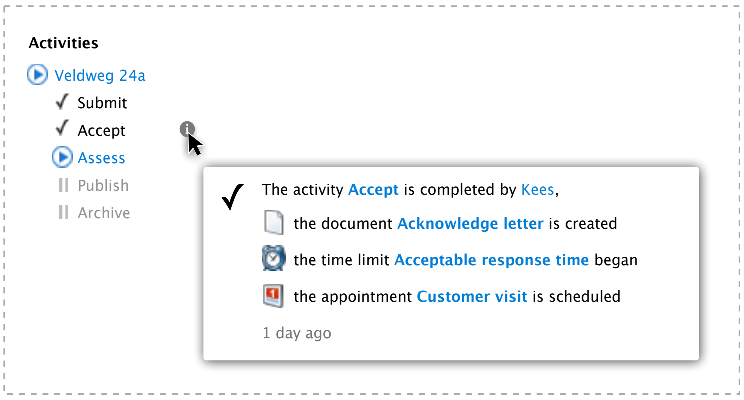
Labels

Postulation completed	The Housing benefit request is completed
Question completed	Is the Housing benefit request completed?
Negation completed	The Housing benefit request is not completed
Postulation in progress	The Housing benefit request is in progre
Question in progress	Is the Housing benefit request in
Negation in progress	The housing benefit requ
Postulation unavailable	Is the H
Question unavailable	
Negation unavailable	

Services

2.4 Textual UI's for Model Driven Applications

Classically, business applications have used tables of data to present detailed information that is available in a business process. When involving customers in business processes, they find it hard to interpret the data. Verbalization into natural language can be a great way to present, for instance, process progress data to laymen, as the data can be presented in a self explanatory way.



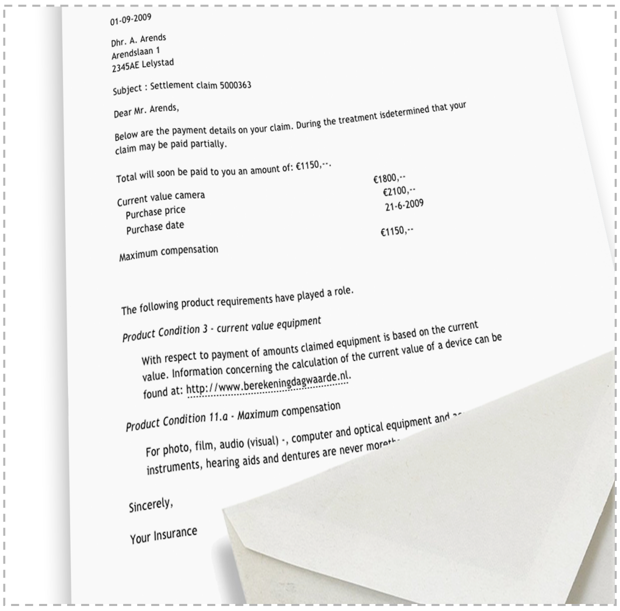
Activities

- ▶ Veldweg 24a
  - ✓ Submit
  - ✓ Accept
  - ▶ Assess
  - ‖ Publish
  - ‖ Archive

✓ The activity Accept is completed by Kees,
the document Acknowledge letter is created
the time limit Acceptable response time began
the appointment Customer visit is scheduled
1 day ago

2.5 Communicating Model Based Decisions

The ontologies capturing legislation and policy are used to drive decision services, applying the policy to actual cases. These decisions taken are communicated to the stakeholders and need to be documented and explained. Verbalization of the model could be extended to verbalization of the decisions based on the models.



01-09-2009

Dhr. A. Arends
Arendslaan 1
2345AE Lelystad

Subject : Settlement claim 5000363

Dear Mr. Arends,

Below are the payment details on your claim. During the treatment isdetermined that your claim may be paid partially.

Total will soon be paid to you an amount of: €1150,--.

Current value camera	€1800,--
Purchase price	€2100,--
Purchase date	21-6-2009
Maximum compensation	€1150,--

The following product requirements have played a role.

Product Condition 3 - current value equipment

With respect to payment of amounts claimed equipment is based on the current value. Information concerning the calculation of the current value of a device can be found at: http://www.berekeningdagwaarde.nl.

Product Condition 11.a - Maximum compensation

For photo, film, audio (visual) -, computer and optical equipment and
instruments, hearing aids and dentures are never more

Sincerely,

Your Insurance

3 Verbalization of Be Informed Models and Meta Models

Within Be Informed, models contain concepts, relations with other concepts, properties and expressions. In general these can be regarded as triples. All are typed and verbalization patterns typically are chosen per type.

3.1 Verbalization of Instances, based on Types

When verbalizing an ontology in many languages, we have to deal with both the types and the instances.

Req. 3.1

When sentences need to be generated for the instances in an ontology, the key lexical/grammatical aspects and sentence structure originate at the type level in the ontology and should be captured at type level.

This is specifically true for verbalization of relations, as the relation instances are typically not named and can only be represented in text by the name of their relation type.

For instance, all pre conditions for an activity are verbalized using a ‘<Activity> may be performed only if <Condition> is met’ pattern. These choices in representation are bound to the pre condition relation and not to the activity or condition type semantics. [CNL2012]

3.2 Verbalization of property values

Properties in Be Informed models can be of a number of types, including numbers, amounts, dates.

Req. 3.2

Property values should be included in localized form in verbalization:

- Numbers and amounts have localized interpunction;
- Dates have appropriate date masks;

Another aspect that could be explored in the project, but is probably out of scope is the verbalization into language form of these properties.

Req. 3.3

Properties could be verbalized into language form:

- Dates could be verbalized: September 22, 1975
- Numbers: one hundred twenty six.

The appreciation by readers is doubtful, or at least depends on the situation. Strategies like verbalizing only whole numbers below hundred for instance could be explored.

A special category of properties is that of expressions, such as numerical formulae. It could be explored whether they could be verbalized into natural language sentences also. There is potential synergy with the Math case here. This one is also considered out of scope of the project, but could be a nice add-on.

Like with the number verbalization, heuristics on when to verbalize and when to show a formula could be explored. In expressions, parenthesis and precedence semantics pose extra challenges when natural language helps and when it gets too complex in natural language also.

3.3 Support for Label Styles

In practice, different styles of choosing labels are found when modelers are, for instance, modeling and naming concepts within a business process model.

Req. 3.4

The sentence generation must, when in-lining instance labels, deal with the grammatical structure of the label and make sure the resulting sentence is correct.

Req. 3.5

The sentence generation must be able to manipulate the label, by for instance changing its verb form, so that its meaning stays the same but it fits better in the resulting sentence.

A preliminary list of label types, also used in [CNL2012], that proved useful in verbalization of conditions and activity models are:

- Names
    - “Intake”, “44b”
- Verb oriented
    - i.e. “Publishing the result”, “Publish the result”
- Propositional
    - “Applicant is over 25”, “Applicant submitted all data”

3.4 Sentence Planning

Typically, model fragments are verbalized into single sentences.

Req. 3.6

The sentence generation must be able to verbalize a single subject with a number of triples that share that subject.

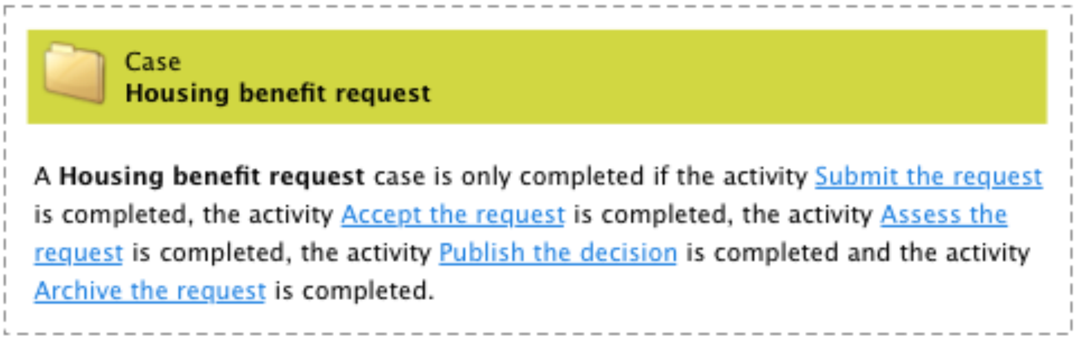
Req. 3.7

The sentence generation should apply sentence planning strategies to make sure that fluent sentences are generated that do not expose unnecessary ontology structure.

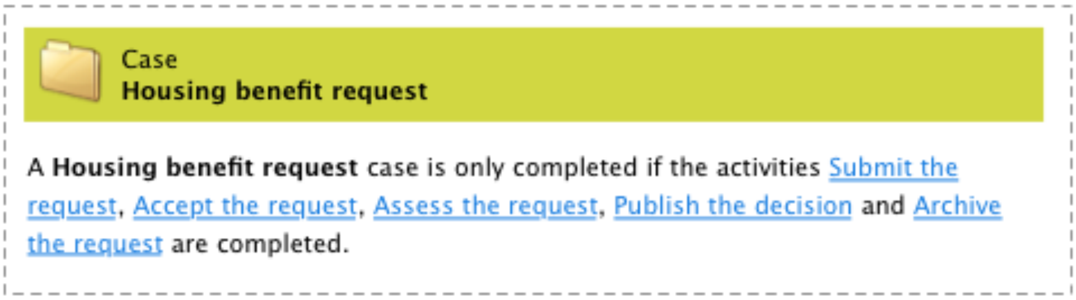
Strategies that are either investigated in the existing prototype or have been identified based on the text it generates are:

- Grouping objects that share a triple type;
    - i.e. “The car is big and the car is yellow” -> “The car is big and yellow”;
- Moving triple verbalization to an adjective
    - i.e. the yellow car is big.

In case of the GF framework, we should investigate at what level to implement these kind of rules: At the Abstract Syntax Tree (AST) level or at serialization level.


Case
Housing benefit request

A **Housing benefit request** case is only completed if the activity Submit the request is completed, the activity Accept the request is completed, the activity Assess the request is completed, the activity Publish the decision is completed and the activity Archive the request is completed.


Case
Housing benefit request

A **Housing benefit request** case is only completed if the activities Submit the request, Accept the request, Assess the request, Publish the decision and Archive the request are completed.

Note that all sentences presented in this version of the document are generated by our prototype and are not representative of the result. For instance, this last sentence should probably be: *'A housing benefit request case is completed if the request is submitted, accepted and assessed, the decision is published and the request is archived.'*

3.5 Different Styles of Concatenation

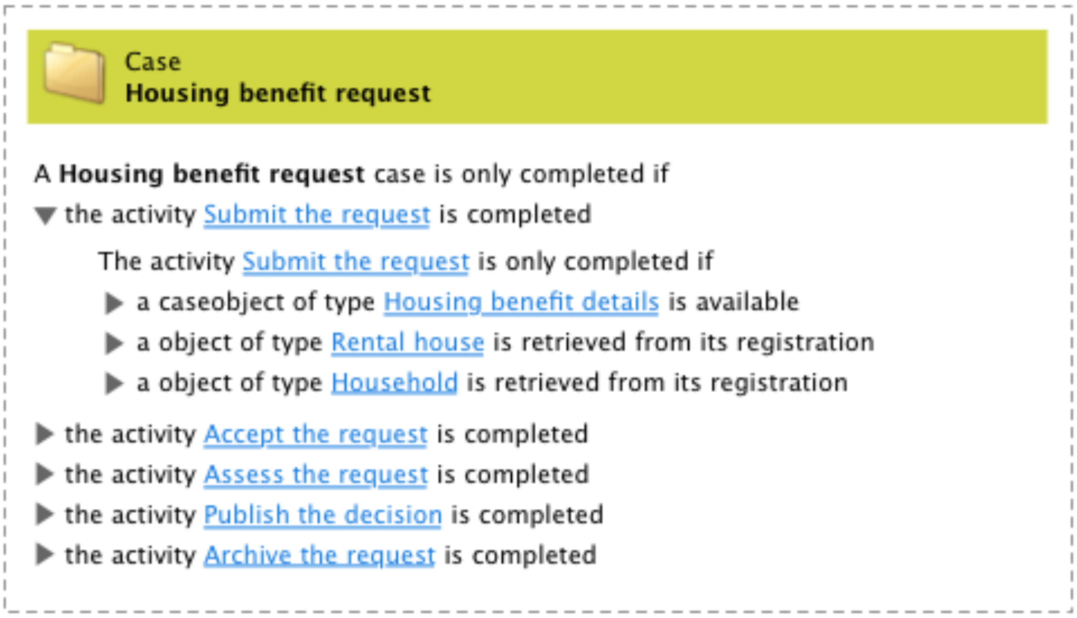
We have experience with different ways of formatting concatenation.

Req. 3.8

The verbalization component should support:

- Using interpunction to combine sub-sentences and lists;
- Using indentation or bullets.

[CNL2010Preproceedings] suggests that domain experts untrained in formal knowledge representation respond differently based on the concatenation strategy.


Case
Housing benefit request

A **Housing benefit request** case is only completed if
▼ the activity Submit the request is completed
The activity Submit the request is only completed if
▶ a caseobject of type Housing benefit details is available
▶ a object of type Rental house is retrieved from its registration
▶ a object of type Household is retrieved from its registration
▶ the activity Accept the request is completed
▶ the activity Assess the request is completed
▶ the activity Publish the decision is completed
▶ the activity Archive the request is completed

3.6 Discourse

When verbalizing a number of concepts or a complete model, a logical order and potentially other forms of emphasis have to be chosen to ensure that a sensible story emerges.

Req. 3.9

Verbalization should support influencing how facts are presented in a logical order.

If used, verbalization should support influencing the use of other forms of emphasis, such as which facts are moved into adjective form in sentence planning.

4 Generating Sentence Variants for Different Tasks

To be useful in a context as Be Informed's, verbalizations will have to take into account the different tasks that are typically performed based on the ontologies and, consequently, are supported by verbalizations.

This chapter describes a number of task contexts in which natural language verbalization of ontologies could be helpful. Although not all are in context of the project, keeping them in mind when grammars are developed could help generalize into these areas later.

4.1 Primary Case: Ontology representation

The primary use case in WP12 is the representation of an ontology for non ontologists to tell them what is expressed in the ontology. Primary applications on this are reviewing and validation when the ontology is created, communicating ontologies when services are based on them.

4.2 Variants needed in Question/Answer Dialog

Typically, when concepts are mentioned in end user dialogs based on ontologies, different forms are used:

- Ask whether a certain concept applies;
    - Are you older than 18?
- State that a concept may/must/may not apply;
    - You must be older than 18!
- State that a certain concept indeed applies;
    - You are older than 18.
- State that is does not apply.
    - You are not older than 18. Or: You are younger than 19.

4.3 Logical Variants for Validation

In review settings, alternative formulation might help in triggering feedback by different kinds of users. The negative formulation or even the contradiction of the statement might trigger more than the formal, positive case. If a user agrees with both a statement and with the opposite, that validation is worthless.

This approach is similar to Terry Halpin's work on ORM verbalization.

For example, the norm on the age of au pairs could be manipulated as follows:

- Every au pair needs to be older than 18.
- No Au pair is younger than 18.
- Au pairs exist that are younger than 18.
- It is not possible that there will ever be an au pair younger than 18

4.4 Feedback & Task Names

In validation and feedback situations, we could trigger feedback by verbalizing possible objections or suggestions a user might have.

- Being older than 18 is not a requirement to be an au pair.
- There are additional requirements to be an au pair.
- Add an extra condition to be an au pair.

5 User Interaction with Generated Text

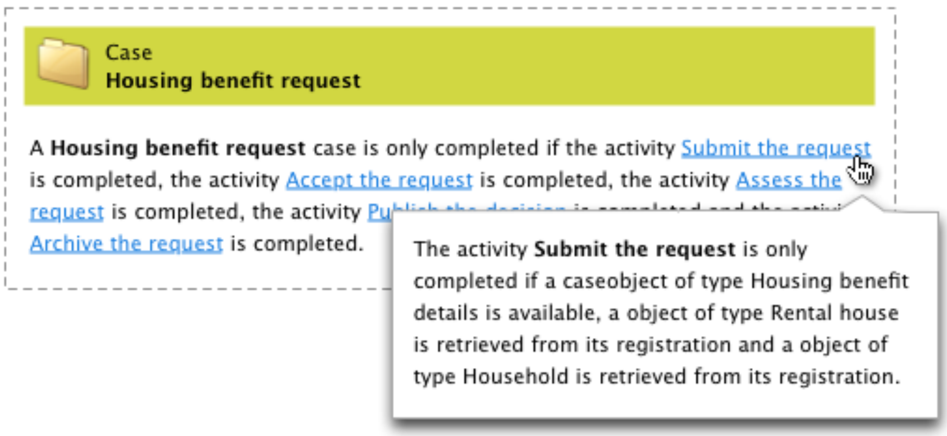
This chapter contains requirements on how the generated texts will be presented to users in different applicative settings and how these users will interact with these texts.

5.1 Online Presentation of Generated Texts

5.1.1 Presenting Annotations

When natural language is used to represent an underlying conceptualization, like an ontology, typically not all of the information in the model is represented in text. Especially in an online environment, there are multiple ways to visualize annotations, also according to the type of information they contain.

- Hyperlinks are typically visualized as underlined fragments of text;
- Details or comments could be visualized as hovering balloons;
- Tags or classifications could be shown using highlighting or inlined visual clues.


Case
Housing benefit request

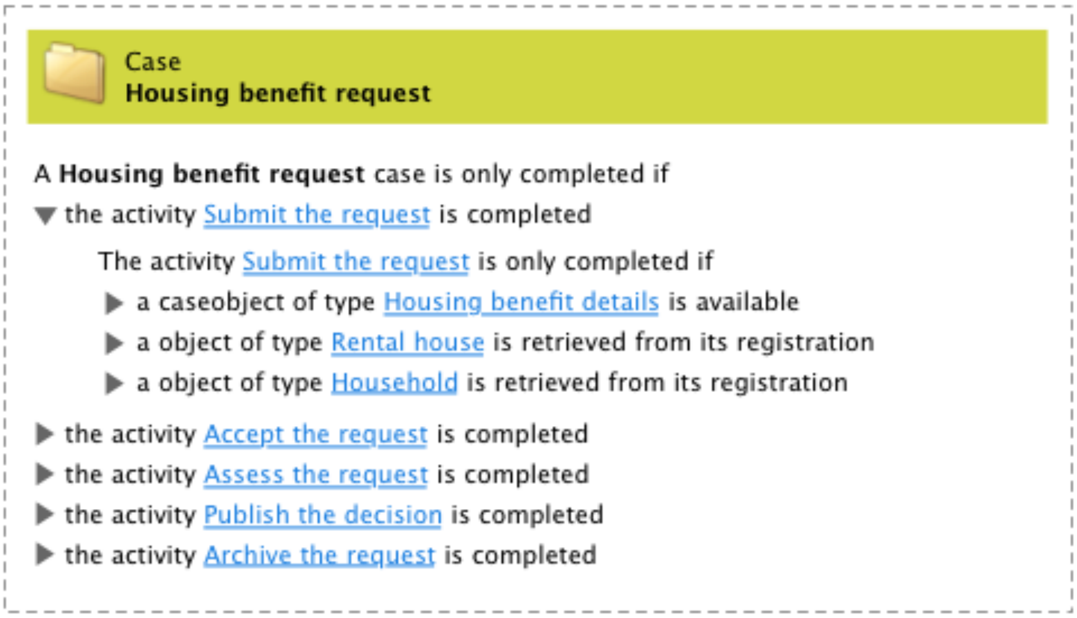
A **Housing benefit request** case is only completed if the activity Submit the request is completed, the activity Accept the request is completed, the activity Assess the request is completed, the activity Pu
Archive the request is completed.

The activity **Submit the request** is only completed if a caseobject of type Housing benefit details is available, a object of type Rental house is retrieved from its registration and a object of type Household is retrieved from its registration.

Technically, this requires having annotations available at the abstract syntax level and on linearization to trace them to the concrete syntax tree and consequent manipulations.

5.1.2 Variable Level of Detail

Verbalization engine should allow readers to influence levels of detail by supporting folding/collapsing or other ways to specify the required level of detail.


Case
Housing benefit request

A **Housing benefit request** case is only completed if

▼ the activity Submit the request is completed

The activity Submit the request is only completed if

▶ a caseobject of type Housing benefit details is available

▶ a object of type Rental house is retrieved from its registration

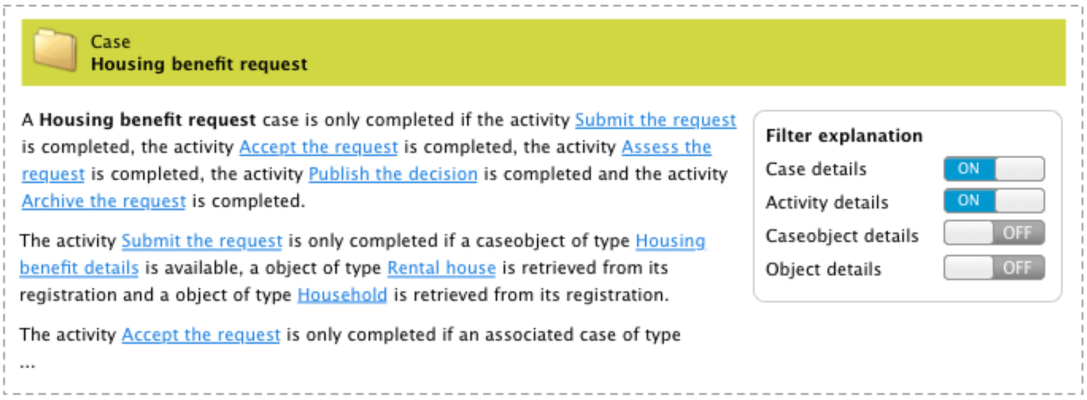
▶ a object of type Household is retrieved from its registration

▶ the activity Accept the request is completed

▶ the activity Assess the request is completed

▶ the activity Publish the decision is completed

▶ the activity Archive the request is completed


Case
Housing benefit request

A **Housing benefit request** case is only completed if the activity Submit the request is completed, the activity Accept the request is completed, the activity Assess the request is completed, the activity Publish the decision is completed and the activity Archive the request is completed.

The activity Submit the request is only completed if a caseobject of type Housing benefit details is available, a object of type Rental house is retrieved from its registration and a object of type Household is retrieved from its registration.

The activity Accept the request is only completed if an associated case of type

...

Filter explanation

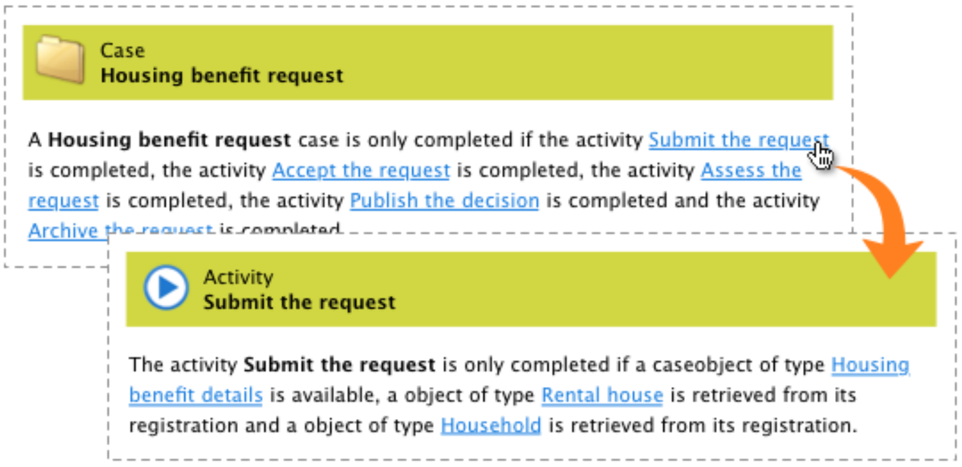
Case details	ON
Activity details	ON
Caseobject details	OFF
Object details	OFF

5.2 Interactive use of Generated Text

This section presents a number of applicative requirements that might apply when generated texts are used in application context.

5.2.1 Support Navigation

It should be possible to support navigation to other application contexts based on the concepts verbalized into a sentence.


Case
Housing benefit request

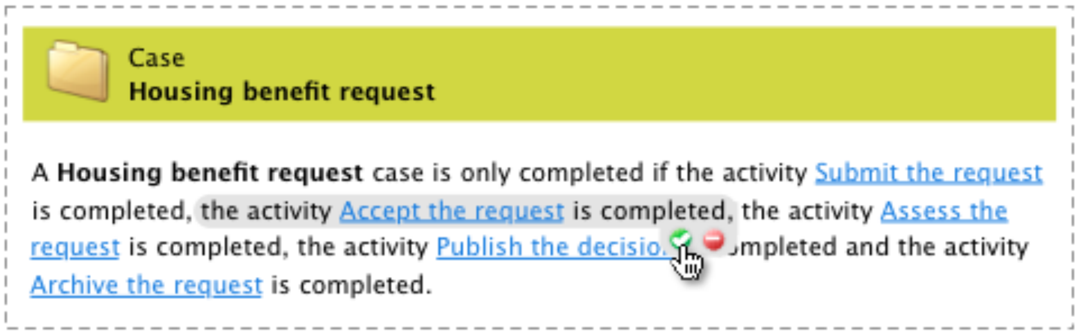
A **Housing benefit request** case is only completed if the activity Submit the request is completed, the activity Accept the request is completed, the activity Assess the request is completed, the activity Publish the decision is completed and the activity Archive the request is completed.

Activity
Submit the request

The activity **Submit the request** is only completed if a caseobject of type Housing benefit details is available, a object of type Rental house is retrieved from its registration and a object of type Household is retrieved from its registration.

5.2.2 *Support editing/manipulation of text*

It should be possible to develop interactive applications based on the textual representation. This requires the support for embedding editing controls and other UI elements that allow users to act upon the text.


Case
Housing benefit request

A **Housing benefit request** case is only completed if the activity Submit the request is completed, the activity Accept the request is completed, the activity Assess the request is completed, the activity Publish the decisio ompleted and the activity Archive the request is completed.

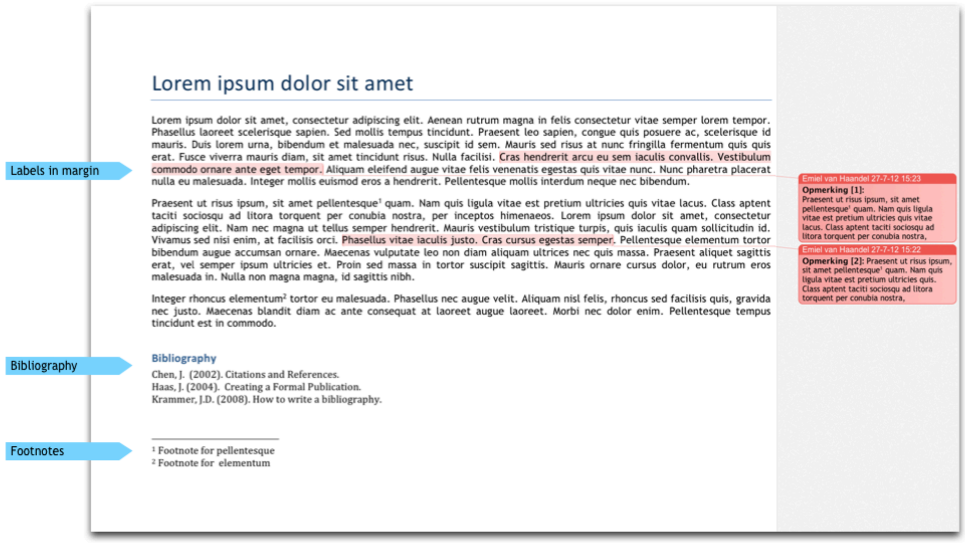
When, as in our scenario, text is just one of a number of visualizations of an underlying conceptualization, these editing operations are most likely performed on the conceptual level, followed by rerendering the text.

Technically, many of these editing operations may be implemented by allowing anchors and regions in the tree and by attaching properties like links, annotations etc to these anchors. Anchors would have to set to abstract tree and propagated through concrete syntax. This could pose challenges when a concrete syntax splits up what is a single node in the AST.

5.3 Requirements for Printed Text

A lot of the requirements posed for applications should be translated to printed text as well, as the same texts are sent in print or are included in static documentation. They need to be mapped to static, document oriented equivalents such as footnotes, parenthesis, font and text formatting, page refs, labels in margin, etc.

Online Requiremen	Document Equivalent
Annotations	Footnotes, Bibliography & References, Labels in margin.
Level of Detail	
Navigation	Table of Contents, in document section references, with page numbers to allow navigation. Indices and glossaries.
Editing	Documents are inherently read only, but form elements could be included.



Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean rutrum magna in felis consectetur vitae semper lorem tempor. Phasellus laoreet scelerisque sapien. Sed mollis tempus tincidunt. Praesent leo sapien, congue quis posuere ac, scelerisque id mauris. Duis lorem urna, bibendum et malesuada nec, suscipit id sem. Mauris sed risus at nunc fringilla fermentum quis quis erat. Fusce viverra mauris diam, sit amet tincidunt risus. Nulla facilisi. Cras hendrerit arcu eu sem iaculis convallis. Vestibulum commodo ornare ante eget tempor. Aliquam eleifend augue vitae felis venenatis egestas quis vitae nunc. Nunc pharetra placerat nulla eu malesuada. Integer mollis euismod eros a hendrerit. Pellentesque mollis interdum neque nec bibendum.

Praesent ut risus ipsum, sit amet pellentesque[1] quam. Nam quis ligula vitae est pretium ultricies quis vitae lacus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam nec magna ut tellus semper hendrerit. Mauris vestibulum tristique turpis, quis iaculis quam sollicitudin id. Vivamus sed nisi enim, at facilisis orci. Phasellus vitae iaculis justo. Cras cursus egestas semper. Pellentesque elementum tortor bibendum augue accumsan ornare. Maecenas vulputate leo non diam aliquam ultrices nec quis massa. Praesent aliquet sagittis erat, vel semper ipsum ultricies et. Proin sed massa in tortor suscipit sagittis. Mauris ornare cursus dolor, eu rutrum eros malesuada in. Nulla non magna magna, id sagittis nibh.

Integer rhoncus elementum[2] tortor eu malesuada. Phasellus nec augue velit. Aliquam nisl felis, rhoncus sed facilisis quis, gravida nec justo. Maecenas blandit diam ac ante consequat at laoreet augue laoreet. Morbi nec dolor enim. Pellentesque tempus tincidunt est in commodo.

Emiel van Haandel 27-7-12 15:23
Opmerking [1]:
Praesent ut risus ipsum, sit amet pellentesque' quam. Nam quis ligula vitae est pretium ultricies quis vitae lacus. Class aptent taciti sociosqu ad litora torquent per conubia nostra,

Emiel van Haandel 27-7-12 15:22
Opmerking [2]: Praesent ut risus ipsum, sit amet pellentesque' quam. Nam quis ligula vitae est pretium ultricies quis. Class aptent taciti sociosqu ad litora torquent per conubia nostra,

Bibliography

Chen, J. (2002). Citations and References.
Haas, J. (2004). Creating a Formal Publication.
Krammer, J.D. (2008). How to write a bibliography.

[1] Footnote for pellentesque
[2] Footnote for elementum

Annotations

6 Embedding Verbalization in the Modeling Process

This chapter will describe how the activities for the development of verbalization related artifacts could fit into the process of business modeling and which roles should typically perform these activities.

This is an area where WP12 should provide useful experiences and that, by nature, cannot be constrained by requirements up front. We do formulate some aims and guiding principles.

6.1 Overall Aim: No Linguistic Development at Modeling Time

An overall requirement, both for the component itself and for process choices how to implement the component, is to reduce or, even better, eliminate the need for linguistically trained personnel, especially at modeling time.

Requiring grammar engineering at modeling time conflicts with two of Be Informed's most important features:

- The ability of business users to exercise ownership over the ontologies, including authoring them. Expecting them to do grammar engineering as part of that role is not realistic;
- The ability to immediately run preview versions of the services and applications inferred from the ontology and use that ability as a tool for validation and feedback in modeling. Requiring grammar engineering as part of changing models would break the immediate nature of that feedback.

6.2 Process Overview

It important to understand the different development processes associated with the Be Informed product.

The product itself is developed at Be Informed R&D, and shipped to our customers and partners. It will have to include:

- Development and maintenance of verbalization component itself;
- Development of grammars for meta models that come with the product.

The standard product is implemented in projects and the implemented product is maintained continuously afterwards. A (very) high overview of the steps in such an implementation is:

- Design
    - High level conceptualization of customers operating models;
    - Optionally meta model extensions if needed to capture customer specific concepts;
- Detail
    - Modeling;

    - Model Translation;
- Develop
    - Develop custom components and integration of solution into landscape;
- Deploy

6.3 Verbalization Related Tasks

- **Modeling** consists of adding concepts and relations to model, and important from a lexical perspective, choosing labels for them;
- **Translation of models** requires providing translations for any natural language element in a model, most notably the labels of course. Typically performed by a combination of professional translators and domain experts. Model translation typically requires familiarity with the domain rather than translation skills, as labels typically contain jargon and the amount of free text is typically low.
- **Translation of model grammars** is implicitly performed when a model is translated. See [CNL 2012].
- **Translating meta model grammars** requires more linguistic knowledge, as meta models may require specific sentence constructs and relations to the resource grammars;
- **Extending meta model grammars** when meta models are extended
- **Creating grammars for new meta models** combines the tasks of extending existing grammars with a design perspective on a grammar.
- **Extending language support**, in GF implemented in resource grammars, is probably the most linguistically demanding task. While writing [CNL 2012] support for the English gerund verb form was extended, which proved to require deep GF expertise. Of course, that is exactly why these resource grammars are developed: To facilitate less experienced grammar developers with support of frequent constructs in concrete languages.

6.4 Typical Roles

The technology and its methodological implications should map well to the (typical) capability profiles of our users.

- Business Modelers/Analysts and Functional Designers are capable of conceptualization of business constraints and policies into ontologies;
- Technical Writers and Translators typically have a language oriented background, but are not always trained in formal linguistic approaches and typically are not very technically inclined;
- Software Engineers typically are fluent in one or more third generation languages like Java, C, Python etc. and the associated tool chains and frameworks. Skills in functional programming are an exception;

- There exists a class of AI/Linguistics oriented Engineers that have their background in AI or Cognitive Studies and are trained in formal or computational linguistics.

A challenge might arise from the requirement to study effective use of natural language, specifically in business oriented domains, and having to implement it in GF, which is based in functional programming and technical in nature.

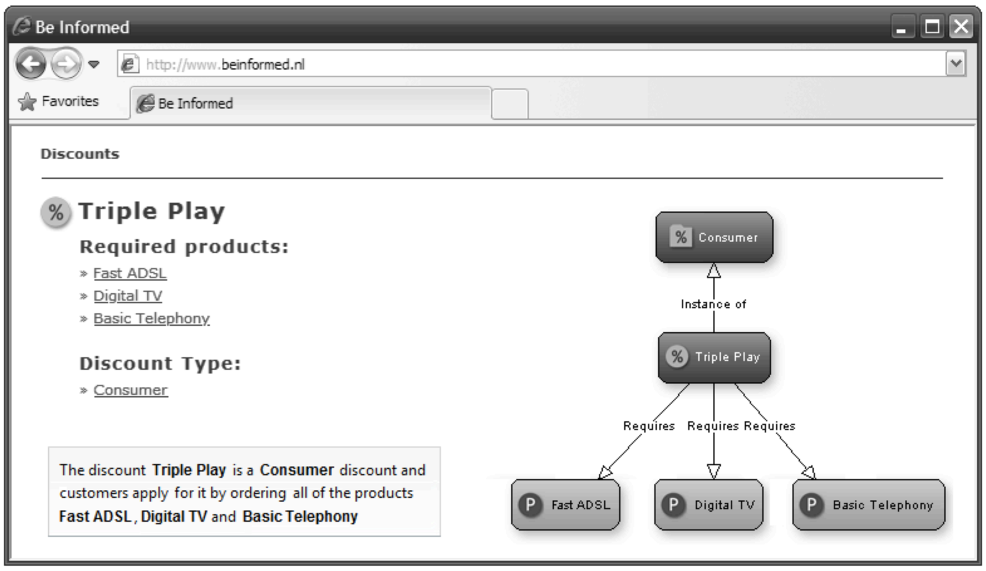
Methodological measures might be needed to separate roles and concerns here into activities that map well on the different roles we currently see in implementation projects.

7 Requirements for Evaluation Purposes

This chapter contains requirements on the verbalization component or any functionality utilizing these verbalizations that deal with the need to evaluate the performance, both of verbalization as a means and the quality/performance of individual grammars.

7.1 Performance of Alternative Visualizations

As natural language is just one of the different visualizations of our models, its relative performance against other graphical or navigational representations of the model is relevant.



The discount **Triple Play** is a **Consumer** discount and customers apply for it by ordering all of the products **Fast ADSL**, **Digital TV** and **Basic Telephony**

Rather than designing experiments, we could also focus on metrics that correlate with performance in actual use of the product in production use.

Req. 7.1

The product should measure the use of the different parallel visualizations in the web interface for navigation.

The relevance of this metric relies on the assumption that there is correlation between the visualization that is preferred/understood/used and the one that is clicked on for navigation.

Req. 7.2

The product should measure the use (number of times, duration, number of edit operations per session) of the different editors that are based on the different visualizations also.

Note that the creation of an editor based on natural language is out of scope of WP12 and so feasibility of this idea is questionable.

Also, the intuition is that the preference of visualization in editing depends strongly on the type of editing task and phase of the project, and the measurement environment should somehow take this into account.

7.2 Quality of Generated Verbalizations

Req. 7.3

Quality of the generated text can be assessed at a number of levels.

7.2.1 Correctness

Check/review (manually) whether the text was grammatically correct.

This would require an export with all verbalizations in a reviewable format.

7.2.2 Appreciation/Subjective Quality

Ask users to rate a text for their appreciation, which should account for softer properties such as fluency. This could be measured by offering a rating functionality or ask users which text of a pair of text alternatives is better.

7.2.3 Performance of the Text

Performance could be measured in certain task context, either by measuring the performance of the task, or asking users whether a certain explanation was helpful.

8 Non Functional Requirements

This chapter contains a number of non-functional requirements that originate from the fact that in WP12 an extension to the Be Informed product is developed.

8.1 Portability

Req. 8.1

The verbalization engine developed needs to be portable across the supported platforms of the Be Informed Product Stack.

That currently consists of the following environments:

- Studio runs on Windows, Mac OSX and Linux.
- Server runs on Windows and Linux.

For precise versions and releases, see our Release Documentation.

Be Informed's portability is based on the use of the Java Runtime Environment.

Platform specific bindings such as spawning processes and native procedure invocation should be prevented whenever possible. If this type of integration is inevitable, it should be cleared with Software Architecture Team before implementing it.

8.2 Modularity

The verbalization engine built should be modular.

Req. 8.2

The engine as a whole should be a drop in replacement of the current verbalization component.

This will allow us to migrate gradually, for instance per domain or per task. Also, it will allow us to show verbalizations of both engines in parallel for evaluation purposes.

Req. 8.3

The integration components developed as part of the verbalization engine should be developed using Be Informed's SDK.

Req. 8.4

The generic GF components used will be used as is, without modifications for integration purposes.

It is important not to create specific versions of GF components for this work package, as it will jeopardize our ability to upgrade in the future.

8.3 Robustness and Graceful Degradation

Req. 8.5

The verbalization engine developed must be robust under unexpected use or operation under suboptimal conditions.

The verbalization engine, to this end, shall exhibit graceful degradation. This means that performance may degrade when conditions worsen, but this degradation should not be abrupt.

This might be a challenge in an inherently rule driven environment such as GF, which by nature have a well-defined domain of validity and do not perform out of domain at all.

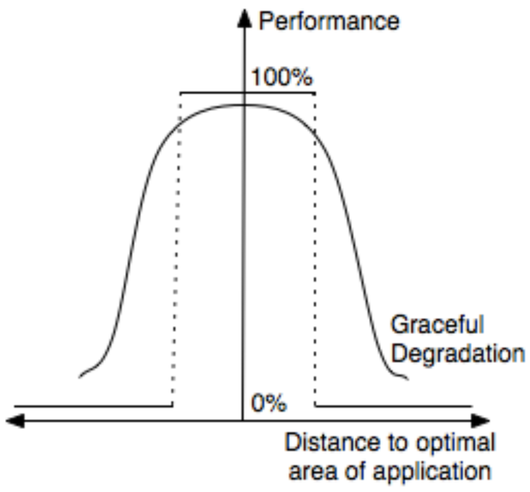


Figure 2: Trade-off between Peak Performance vs. Out-of-lab Performance

An example of an approach to address this requirement is having parallel rule sets: One that leverages specific knowledge and gives great results, and one that has less prerequisites and is broadly applicable but degrades is performance.

8.4 Legal

Req. 8.6

We must make non-infectious use of GF technology as allowed by their associated open source licenses.

We can integrate at code level and/or embed in our installers any open source component released under, among others, Apache 2.0, EPL, MIT style and LGPL.

We can probably integrate over open system-to-system interfaces with software components released under other, more restrictive, licenses such as GPL. We cannot package these with our software. This needs to be reviewed by legal at a case by case level.

9 Bibliography

[ESWC2009] Heller, R., & Van Teeseling, F. (2009). Knowledge Applications for Life Events: How the Dutch Government Informs the Public about Rights and Duties in the Netherlands. In L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, et al. (Eds.), *The Semantic Web Research and Applications* (Vol. 5554, pp. 846-850). Springer.

[CNL 2010 Preproceedings] Spreeuwenberg, S., Van Grondelle, J., Heller, R., & Grijzen, G. (2010). Design of a CNL to Involve Domain Experts in Modelling. In M. Rosner & N. E. Fuchs (Eds.), *CNL 2010 Second Workshop on Controlled Natural Languages*. CEUR Workshop Proceedings (CEUR-WS.org).

[CNL2010] Spreeuwenberg, S., van Grondelle, J., Heller, R., & Grijzen, G. (2012). Using CNL Techniques and Pattern Sentences to Involve Domain Experts in Modeling. In Mike Rosner & N. E. Fuchs (Eds.), *Proceedings of Controlled Natural Language 2010*. Springer Verlag.

[PoEM2011] Van Grondelle, J. C., & Gulpers, M. (2011). Specifying Flexible Business Processes using Pre and Post Conditions. In P. Johannesson, J. Krogstie, & A. L. Opdahl (Eds.), *Practice of Enterprise Modeling* (Vol. 92, pp. 1-14). Springer.

[EKAW2010] Van Grondelle, J., Heller, R., Van Haandel, E., & Verburg, T. (2010). Involving Business Users in Formal Modelling using Natural Language Pattern Sentences. *EKAW 2010 17th International Conference on Knowledge Engineering and Knowledge Management*. Springer.

[CNL2012] Davis, B., Enache, R., van Grondelle, J., & Pretorius, L. (2012). Multilingual Verbalisation of Modular Ontologies using GF and lemon. *Proceedings of CNL 2012*.

[ESWC2012] van Grondelle, J. (2012). New Audiences for Ontologies : Dealing with Complexity in Business Processes. In E. Simperl, P. Cimiano, A. Polleres, O. Corcho, & V. Presutti (Eds.), *The Semantic Web: Research and Applications, Proceedings of ESWC 2012* (p. 2). Crete, Greece: Springer Verlag.

Wapenrustlaan 11-31
7321 DL Apeldoorn
The Netherlands

T +31 (0)55 368 14 20
E info@beinformed.com

www.beinformed.com

About Be Informed
Be Informed is an internationally operating, independent software vendor. The Be Informed business process platform supports administrative processes, which are becoming increasingly knowledge-intensive. Thanks to Be Informed's unique approach to dynamic case management, the next wave after business process management, organizations using Be Informed often report cost savings of tens of percents. Further benefits include a much higher straight-through processing rate leading to vastly improved productivity, and a reduction in time-to-change from months to days.