Specification

Project Acronym:	PEPPOL
Grant Agreement number:	224974
Project Title:	Pan-European Public Procurement Online

Virtual Company Dossier
Use cases and other UML representations of the European VCD system

Revision: 1.01

Authors:

Ansgar Mondorf (UKL)
Maria Wimmer (UKL)

Project co-funded by the European Commission within the ICT Policy Support Programme

Dissemination Level		
P	Public	X
C	Confidential, only for members of the consortium and the Commission Services	

Revision History

Revision	Date	Author	Organisation	Description
1.0	20100430	Maria Wimmer	UKL	First version (pending EC approval)
1.01	20101001	Klaus Vilstrup Pedersen	DIFI	EC Approved

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Statement of copyright

Contributors

Organisations

ADETEF (Assistance au developpement des echanges en techologies economiques et financieres), France

CNIPA (Centro Nazionale per l'Informatica nella Pubblica Amministrazione[1]) , Italy, www.cnipa.gov.it

DIFI (Direktoratet for forvaltning og IKT[2]), Norway, www.difi.no

EKEVYL (The Research Center for Biomaterials), Greece, www.ekevyl.gr

IC (InfoCamere), Italy, www.infocamere.it

PEPPOL.AT, Austria

UKL (University of Koblenz-Landau), Germany

Persons

Ansgar Mondorf, UKL (editor)
Bruno Boutteau, ADETEF
Bruno Deschemps, ADETEF, France
Daniel Schmidt, UKL
Doris Ipsmiller, PEPPOL.AT
Dörthe Körner, DIFI
Elisabeth Sundholm, DIFI
Ellen Lücke, Beschaffungsamt, Germany
Jean-Philippe Nadal, ADETEF
Josef Makolm, PEPPOL.AT
Lefteris Leontaridis, EKEVYL
Maria A. Wimmer, UKL
Markus Müller, UKL
Markus Schett, PEPPOL.AT
Paola Fumiani, IC
Piero Milani, IC
Sverre Bauck, DIFI
Trygve Laake, DIFI
Wolfgang Groiß, PEPPOL.AT

[1] From 29th December 2009, CNIPA will be renamed DigitPA (Legislative Decree 1st December 2009, n. 177)
[2] English: Agency for Public Management and eGovernment

Table of Contents

1. Use Cases

This chapter provides an overview over the main use-cases and the involved asserted as well as the expected inferred data mainly in order to demonstrate, which concepts are responsible to structure which kind of information.

Ontology Maintenance

Maintenance of EU Criteria: Instances of the classes CriterionGroup and EUCriterion are provided/changed.

Maintenance of National Criteria: Instances of the classes CriterionGroup, NationalAtomicCriterion and NationalCombinedCriterion are provided/changed.

Maintenance of Evidences and Evidence Restrictions: Instances of the classes NationalEvidence and EvidenceRestriction are provided/changed. The Tenderer Schema may be extended to allow specification of certain kinds of Tenderers for Evidence Restrictions.

Maintenance of Criteria Requirements: Instances of the class CriterionRequirement are provided/changed. The tenderer-schema may be extended to allow specification of certain kinds of Tenderers for Criterion Requirements.

Generation of Required Criteria for a provided Tenderer Structure

Instances for classes from the tenderer-schema are provided.
The national criteria for the countries of the tenderer and the contracting authority, the national evidences and evidence restrictions for the country of the tenderer and the criteria requirements for the country of the contracting authority are loaded as well as the EU criteria and any commonly needed ontology parts. For each instance provided in the first step, a list of required criteria in the country of the contracting authority as specified by the criteria requirements is inferred.

Deriving Criteria from Contracting-Authority-National Evidences

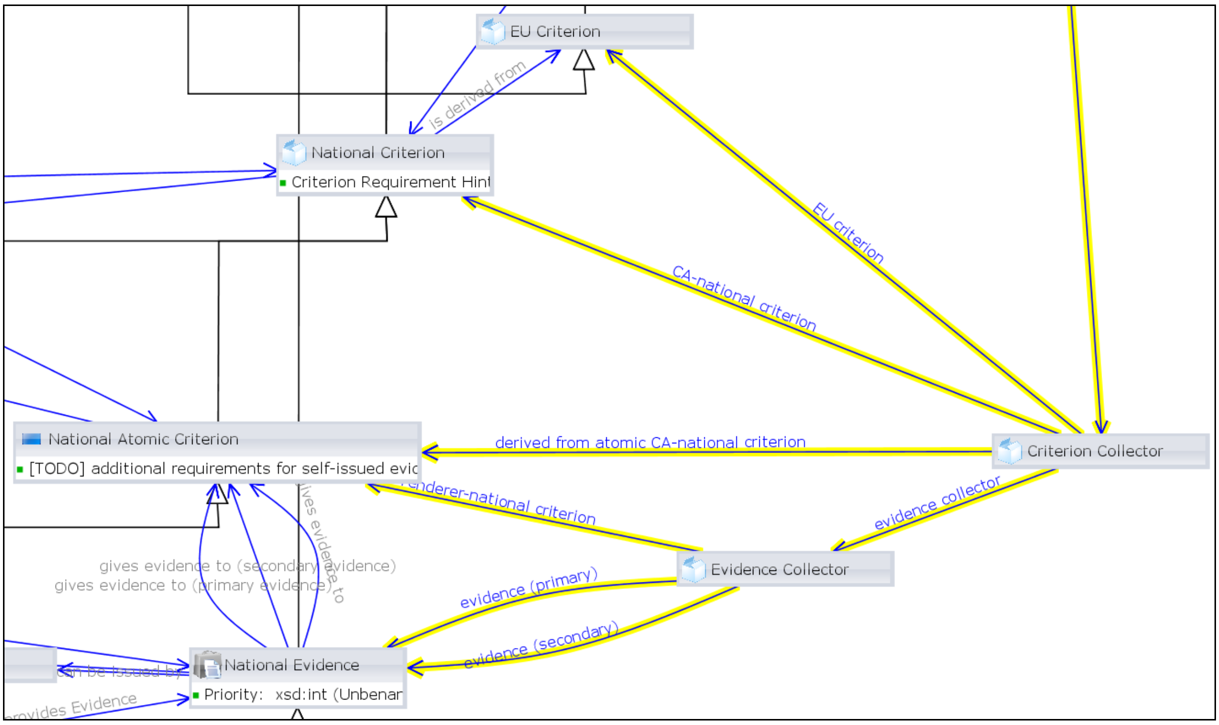
While this has come up as a potential use case for the reasoning, this will not be handled by the reasoner. Instead it is suggested, that the Evidences can be used to let the user search for Criteria in the interaction step.

Generation of possible Evidences for a provided Tenderer Structure with required Criteria

Instances for classes from the Tenderer Schema are provided as well as for each of those instances the criteria (in the country of the contracting authority) intended to be proven.
The national criteria for the countries of the tenderer and the contracting authority and the national evidences and evidence restrictions for the country of the tenderer are loaded as well as the EU criteria and any commonly needed ontology parts.
For each criterion provided in the first step, a list of possible chains of CA-national-criterion → EU-criterion → T-national-criterion → Evidence for each of the evidences suitable for proving the criterion and available to the Tenderer Structure Element is inferred.

1.1 Collector Schema

The Collector Schema as depicted in “Figure 1-1: The Collector Schema” provides the classes and properties used as input and output for the reasoning deriving Evidences from required Criteria.



EU Criterion
is derived from
National Criterion
Criterion Requirement Hint
EU criterion
CA-national criterion
National Atomic Criterion
[TODO] additional requirements for self-issued evic
derived from atomic CA-national criterion
Criterion Collector
enderer-national criterion
evidence collector
gives evidence to
gives evidence to (secondary evidence)
gives evidence to (primary evidence)
Evidence Collector
evidence (primary)
evidence (secondary)
National Evidence
Priority: xsd:int (Unbenar
can be issued by
provides Evidence

Figure 1-1: The Collector Schema

The input for the reasoning is specified by linking the Tenderer Structure Subjects to the criteria to be proven with the “wantsToProveCANC” property.
The output of the reasoning will be grouped into called “Collectors”. Collectors are structured so that they show the actual path from Contracting-Authority-National Criteria via EU Criteria to Tenderer-National Criteria and, as a result, Evidences.
The steps of the reasoning process for deriving Evidences from Contracting-Authority-National Criteria are:

A Criterion Collector for every wantsToProveCANC relationship is created. Atomic Criteria belonging to the same combined Criterion are grouped into a single Collector.

The Contracting-Authority-National Criterionto be proven is added to that Collector, along with the corresponding EU Criterion.

For every Tenderer-National.Atomic Criterion mapped (directly or via combined Criteria) to that EU Criterion, an Evidence Collector is added to the Criterion Collector, with the Tenderer-national-Atomic Criterion attached.

Finally, all evidences suitable to prove that Criterion (respecting Evidence Restrictions) are added to the Evidence Collector.

1.2 Reasoning Workflow

The process of inferring new data out of existing information on top of the rule set is referred to as reasoning. In order to meet all requirements (simplicity, efficient change processes, etc), the reasoning within the VCD European System will have to be split up into several steps. Between some of the reasoning steps, the user is expected to modify the data (represented in RDFS), for example to add/remove Criteria to be proven or to confirm the suggestion made by the system (specifically the reasoners). This interaction is not part of the reasoner itself.

By carefully choosing the sets of schemas and data loaded and considered at each reasoning step, the limitations with regard to OWL DL reasoning can be overcome without losing the advantages of OWL DL: if no data and rules are loaded that violate OWL DL (specifically, the tenderer-criterion-schema and its instances) at the respective step, OWL DL reasoning on the tenderer data can still be performed, the output of that reasoning step can be captured and used as input for the successive reasoning steps.
This workflow is outlined in Figure 1-2: The reasoning workflow. The illustration shows the process (third swim lane) involving user interactions and reasoning steps: the second, third and fifth step are performed by the reasoners while the first and third step will be performed by the user and controlled by some other component.

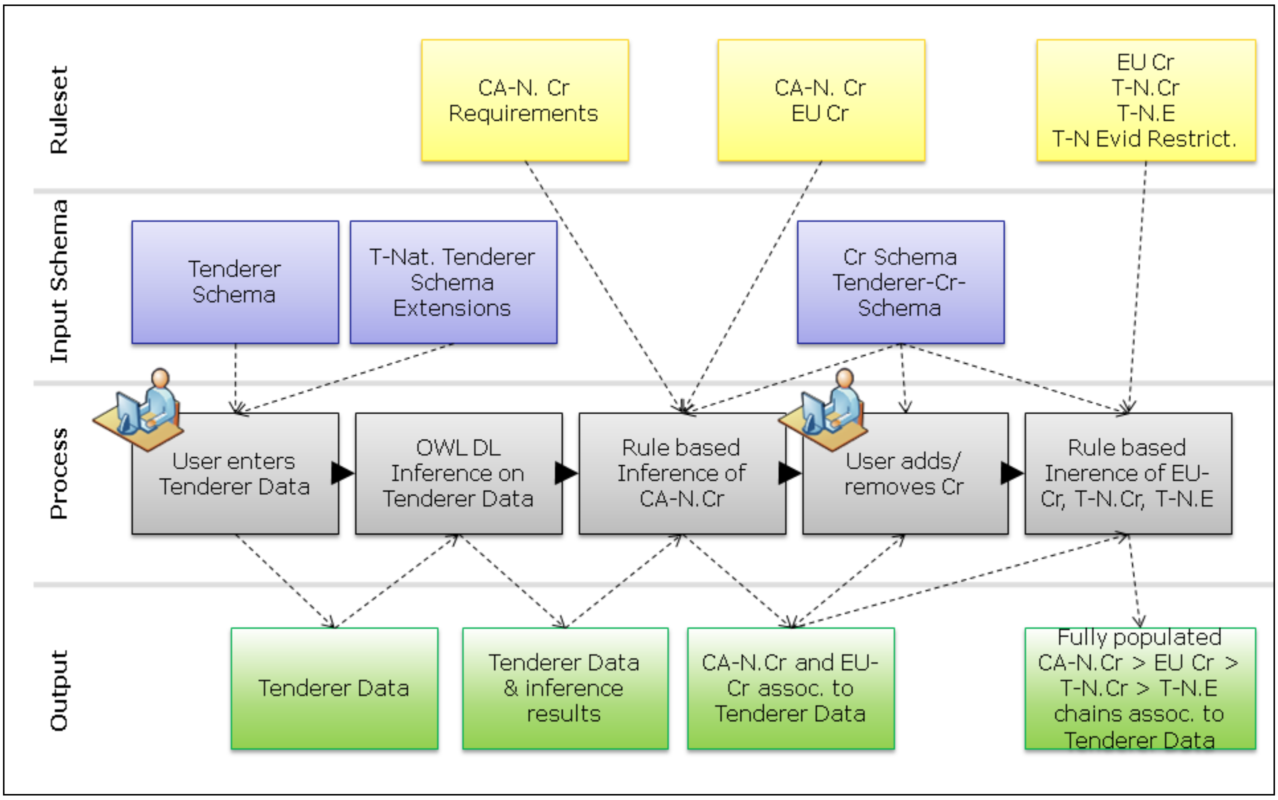

Ruleset
Input Schema
Process
Output
CA-N. Cr Requirements
CA-N. Cr EU Cr
EU Cr T-N.Cr T-N.E T-N Evid Restrict.
Tenderer Schema
T-Nat. Tenderer Schema Extensions
Cr Schema Tenderer-Cr-Schema
User enters Tenderer Data
OWL DL Inference on Tenderer Data
Rule based Inference of CA-N.Cr
User adds/ removes Cr
Rule based Inerence of EU-Cr, T-N.Cr, T-N.E
Tenderer Data
Tenderer Data & inference results
CA-N.Cr and EU-Cr assoc. to Tenderer Data
Fully populated CA-N.Cr > EU Cr > T-N.Cr > T-N.E chains assoc. to Tenderer Data

Figure 1-2: The reasoning workflow

1.3 Inference Rules

We assume OWL DL reasoning has already been performed on top of the tenderer data and the results (especially any additional type statements for instances of the tenderer-schema classes) are available as input to the rules described here.

- Rule Syntax

This chapter describes the syntax for the rules. The syntax isn't specified formally yet and this description of the syntax can only serve as a guideline to help the reader understand the rules shown later on, not as a formal specification of the rule semantics.
Rules consist of preconditions and results, with the preconditions on the left, then a => and finally the results. Preconditions and results are sets of triple patterns. A single triple pattern is denoted as (<s> <p> <o>), where <s> denotes the subject, <p> the predicate and <o> the object of the RDF triple. A predicate of a is shorthand for rdf:type.
Each element of a triple can be either an URI node, a variable node or a creation-variable node.
URI nodes are denoted as simple strings, variable nodes are strings prefixed with a ? and creation-variable nodes are strings prefixed with a $.
URI nodes can appear anywhere in a rule. Variable nodes can only appear in the results of a rule if they have appeared in the preconditions of that rule. Creation-Variable nodes can only appear in the results of a rule.
Rules are applied by trying to bind variable nodes in the preconditions to URI nodes in the base asserted data so that the preconditions are all satisfied. Variable bindings from one triple in the preconditions of a rule carry over to all other triples in that rule. When all preconditions are met, the triple patterns in the results – with the variable nodes replaced according to the variable bindings in the matching step – are emitted. When emitting, each creation-variable node in the results triple patterns is replaced by a single newly-created URI node.
Apart from triple patterns, results can also contain nested rules that become active when the preconditions of the outer rule are met. A nested rule can include an ELSE clause that will be used to produce triples if the preconditions of the nested rule are NOT met (but the preconditions of the outer rule are met). Variable bindings from the outer rule carry over to the nested rule.
Nested rules can be denoted in [] or by giving them a name in caps with variables to be carried over as parameters in ().

- Deriving suggested criteria from criteria requirements (see 0)

(?x a ?c) (?req a CriterionRequirement) (?req tendererType ?c) (?req requires ?crit) => (?x wantsToProveCANC ?crit)

- Deriving Tenderer-National Criteria from Contracting-Authority-National Criteria

Input for this step would be either the (possibly interactively modified) results or data provided by the tenderer himself expressed in the same schema.

Building the criterion collectors (reducing atomic to combined CA-national criteria if possible)
(?x wantsToProveCANC ?canCrit) => [
 (?canCrit a AtomicCriterion) (?canCombined consistsOf ?canCrit) => CRIT_COLLECTOR(?x, ?canCombined, CriterionCollector, criterionCollector, canCriterion)
 ELSE CRIT_COLLECTOR(?x, ?canCrit, CriterionCollector, criterionCollector, canCriterion)
]
COLLECTOR(?subject, ?criterium, ?type, ?subjectToCollector, ?collectorToCriterium): (?subject ?subjectToCollector ?alreadyExistingCollector) (?alreadyExistingCollector ?collectorToCriterium ?criterium) => () ELSE (?x ?subjectToCollector $collector) ($collector ?collectorToCriterium ?criterium) ($collector a ?type)
(?x wantsToProveCANC ?canCrit) (?canCrit a AtomicCriterion) (?canCombined consistsOf ?canCrit) (?x criterionCollector ?cc) (?cc canCriterion ?canCombined) => (?cc derivedFromAtomicCANCriterion ?canCrit)

Deriving TN-Criteria via EU-Criteria (reducing combined T-National Criteria to atomic) and building the evidence collectors
(?cc canCriterium ?canCrit) (?canCrit isDerivedFrom ?euCrit) (?tnCrit isDerivedFrom ?euCrit) => (?cc euCriterium ?euCrit) [
 (?tnCrit a CombinedCriterion) (?tnCrit consistsOf ?tnSub) => COLLECTOR(?cc, ?tnSub, EvidenceCollector, evidenceCollector, tnCriterion)
 ELSE COLLECTOR(?cc, ?tnCrit, EvidenceCollector, evidenceCollector, tnCriterion)

```
  (?canCrit minimumSubstitutionLevel ?sl) => (?cc minimumSubstitutionLevel ?sl)
]
```

- Deriving Evidences from Tenderer-National Criteria

```
(?tse criterionCollector ?cc) (?cc evidenceCollector ?ec) (?ec tnCriterion ?tnc) (?ev givesEvidenceTo
?tnc) =>[
  (?er a EvidenceRestriction) (?er evidenceOfType ?ev) => RESTRICTED(?tse, ?ec, ?ev, ?er)
  ELSE (?ec evidence ?ev)
]
RESTRICTED(?tse, ?collector, ?ev, ?er): (?er isAvailableFor ?tType) (?tse a ?tType) => (?collector
evidence ?ev)
  (?cc minimumSubstitutionLevel ?sl) => (?ec minimumSubstitutionLevel ?sl)
```

1.4 (simplified) Example of the Reasoning Process

This section gives an example of how the reasoning process is carried out. ”Figure 1-3: An example of the reasoning process” shows the process steps on top, and a simplified illustration of the input and output data of the process steps below; olor coding ties the process steps to their output.
It is assumed, that the Economic Operator acting as a tenderer is Austrian, and the Contracting Authority Italian.

(1) The process begins with the user providing the tenderer structure (gray): A single tenderer, m2n, which is also a legal entity and has a business register number 183286p, and a legal representative, Doris, who is a natural person.

(2) Taking this data as input, OWL DL reasoning is performed (yellow), yielding additional types: m2n must be a registered business because it has a business register number. Doris must be a representative (rdf:type) because it occurs as the object of the „has representative“ property.
Due to clarity in the example illustration this step and how those rules are encoded in the austrian tenderer-schema extensions using OWL DL constructs are not shown.

(3) The result of this step (gray and yellow) is now input for the first rule-based inference step. This results in the creation of criterion collectors (blue). The output of this step is shown in blue.

(4) In the next step, the user is in interacting: parts of the results from the previous step (shown in blue) possibly being removed, and similar parts being added. For simplicity's sake, we assume in the example below, that the user was perfectly satisfied with the results and didn't change the suggested criteria.

5) The last step now performs the reasoning, resulting in Evidence Collectors being tied to the tenderer-national criteria (green) and finally the evidences (light green). The illustration does not show, that „Business Register Extract“ is tied via an evidence restriction to the class „Reg. Business (.at)“, and that „Criminal Record Extract“ is tied to „Natural Person“. Those evidence restrictions are satisfied by the tenderer data, and therefore the evidences are included in the evidence collectors. Any potential evidences whose evidence restrictions are not satisfied will be excluded from the evidence collectors and are hence not shown (e.g. a special kind of criminal record extract only available for companies and not for natural persons).

Evidence collector 1 illustrates that this process can result in multiple evidences for a single evidence collector, implying that the user is free to choose one of the evidences (but might be bound by their relative priorities and their primary/secondaryness). That choice is outside of the scope of the reasoning process described here but will probably governed and controlled by the same component that controls this process.

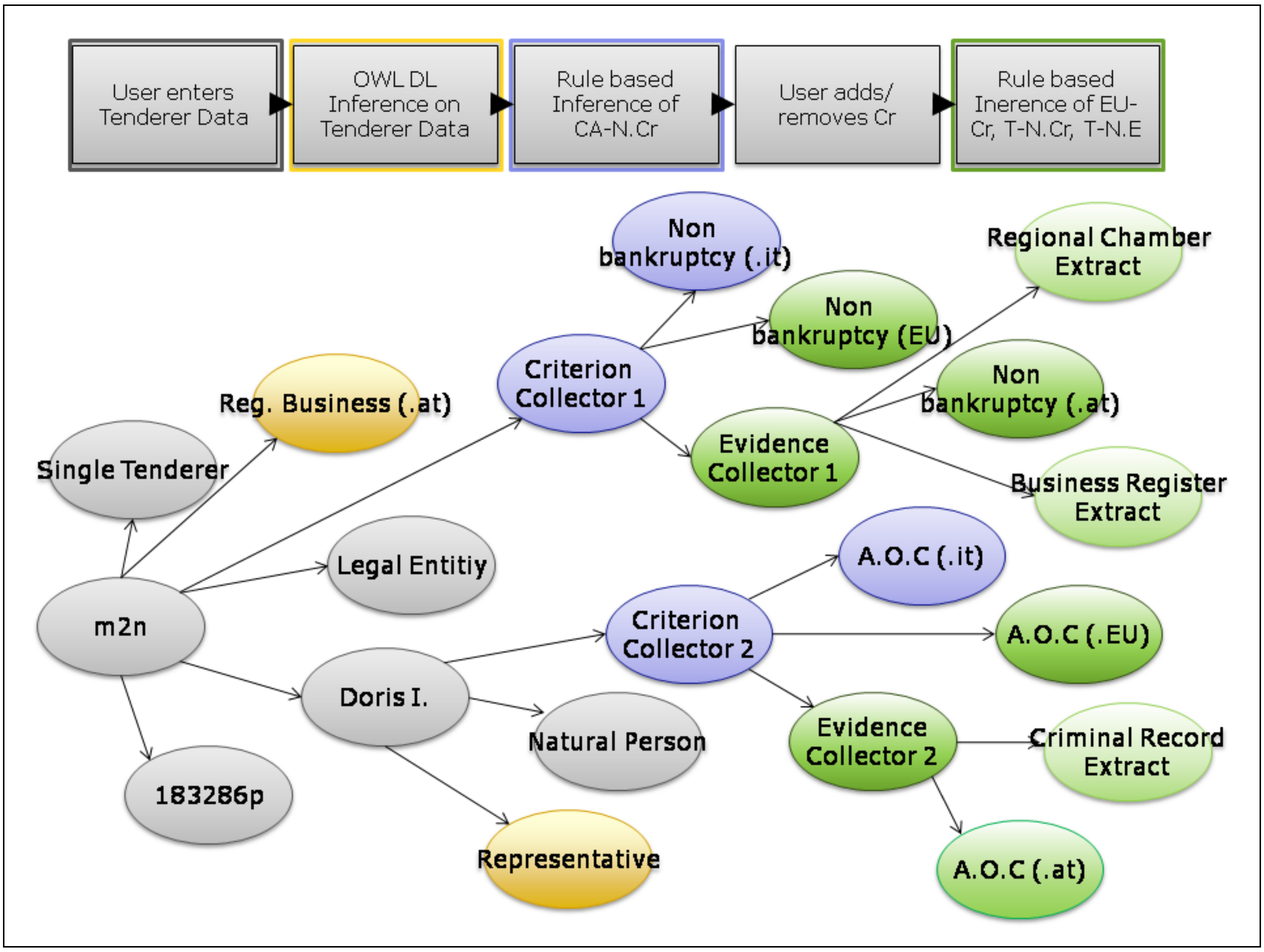


Figure 1-3: An example of the reasoning process

2 Functional specification of VCD Ontology Maintenance Tool

The *VCD Ontology Maintenance Tool* of the European VCD System is providing functionality to generate, modify and administrate the VCD Ontology which capturing the relevant information for deriving the Evidences[3].

2.1 Use Case “VCD Ontology Maintenance Tool”

The following diagram shows the use cases of the VCD Ontology Maintenance Tool.

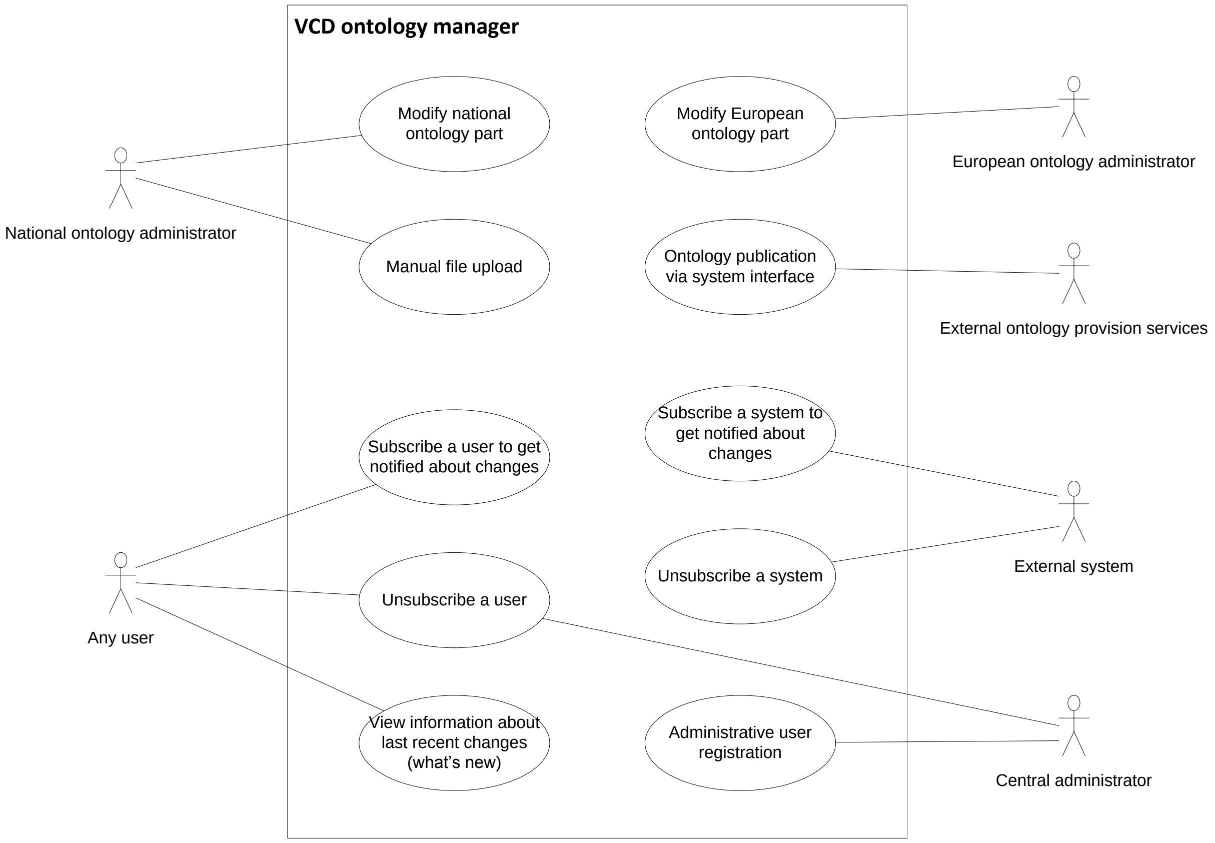

VCD ontology manager
Modify national ontology part
Modify European ontology part
European ontology administrator
National ontology administrator
Manual file upload
Ontology publication via system interface
External ontology provision services
Subscribe a system to get notified about changes
Subscribe a user to get notified about changes
External system
Unsubscribe a system
Any user
Unsubscribe a user
View information about last recent changes (what’s new)
Administrative user registration
Central administrator

Figure 2-1: Use-Cases of the VCD ontology manager (all)

[3] Ref to the specification document to be added

2.2 Use-Case group “Manual Ontology Maintenance”

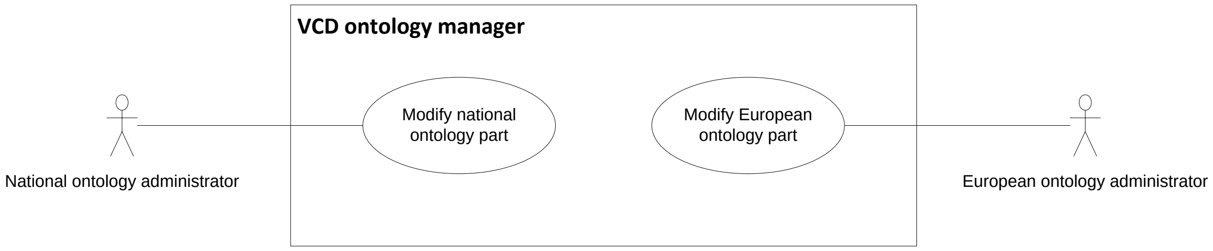

VCD ontology manager
Modify national ontology part
Modify European ontology part
National ontology administrator
European ontology administrator

Figure 2-2: Use-Cases concerning manual Ontology maintenance

Aspect	Description
Objective	An authorized user wants to modify national parts of the ontology.
Results (postconditions in case of success)	User is logged into the system National part of the ontology is valid
Precondition	User is logged into the system
Postcondition in case of failure	User is logged into the system National part of the ontology is valid
Actor(s)	National ontology administrator
Initiating event	User wants to modify the national part of the rule set
Description of interaction procedure with European VCD System (standard run)	The user navigates to his national subset of the ontology. The user modifies a part of the ontology, provides a reason for change/comment and saves the changes. The system only allows valid changes, which are not in conflict with the ontology. A modification can be: Adding a new triple (or sets of triples) Altering an existing triple (or sets of triples) Deleting an existing triple (or sets of triples) The system must inform the user about the relevant affected concepts of the ontology (with regard to 0).
Description of interaction procedure with European VCD System (alternative runs)	None
Extension(s)	None

Table 2-1: Modify national ontology part

The activity diagram in Figure 2-3 depicts the scenario described above.

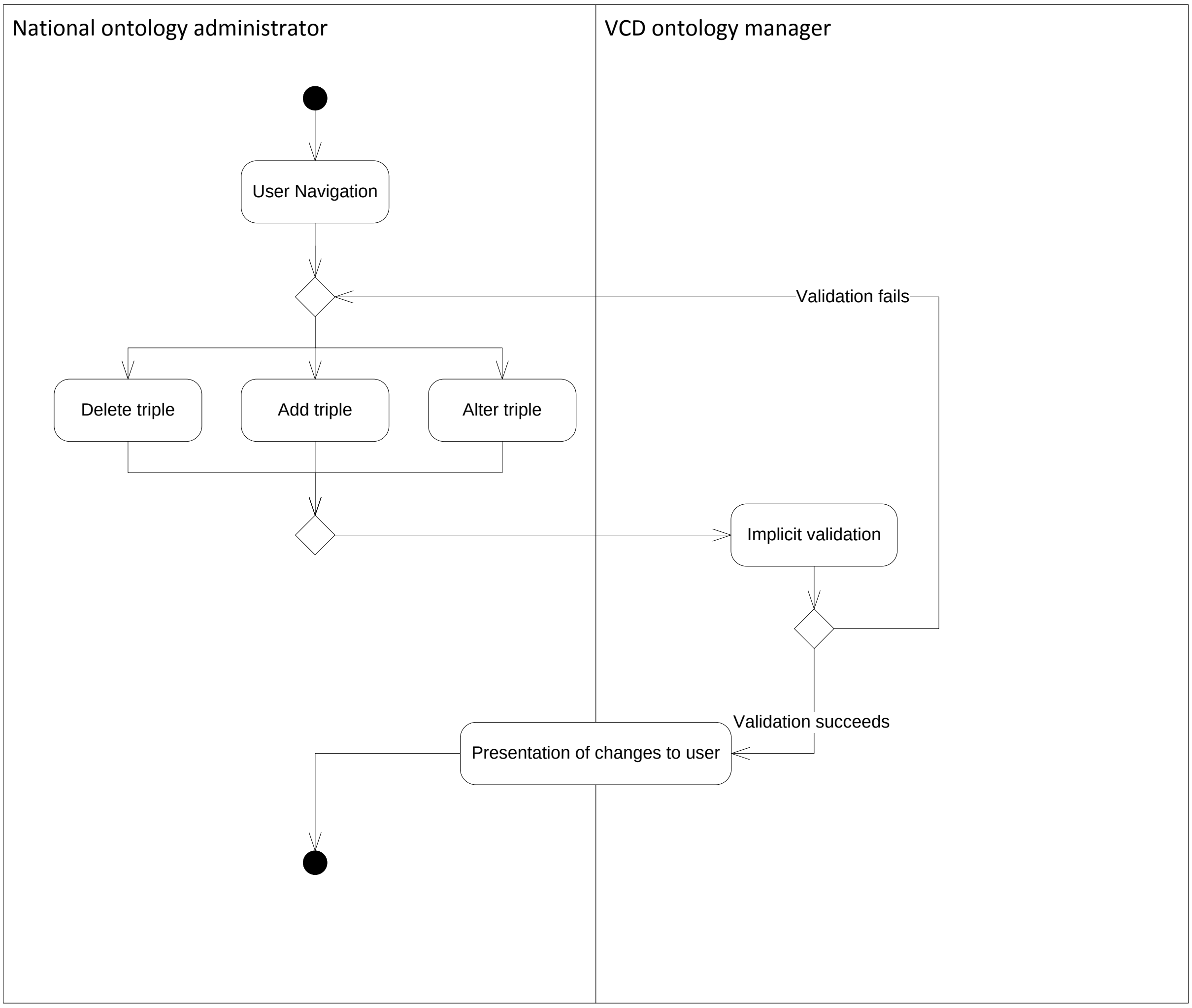


Figure 2-3: Activity diagram for the use case "Modify national ontology part"

Aspect	Description
Objective	An authorized user wants to modify European parts of the ontology.
Results (postconditions in case of success)	User is logged into the system All European parts of the ontology are valid
Precondition	User is logged into the system Governance process has been performed in advance
Postcondition in case of failure	User is logged into the system All European parts of the ontology are valid
Actor(s)	European ontology administrator
Initiating event	User wants to modify European parts of the rule set
Description of interaction procedure with European VCD System (standard run)	The user navigates to the European subset of the ontology, he wants to change. The user modifies a part of the ontology, provides a reason for change/comment and saves the changes. The system only allows valid changes, which are not in conflict with the ontology. The system must inform the user about the relevant affected concepts of the ontology (with regard to 0).
Description of interaction procedure with European VCD System (alternative runs)	None

Extension(s)	None

Table 2-2: Modify European ontology part

The activity diagram in Figure 2-4 depicts the scenario described above.

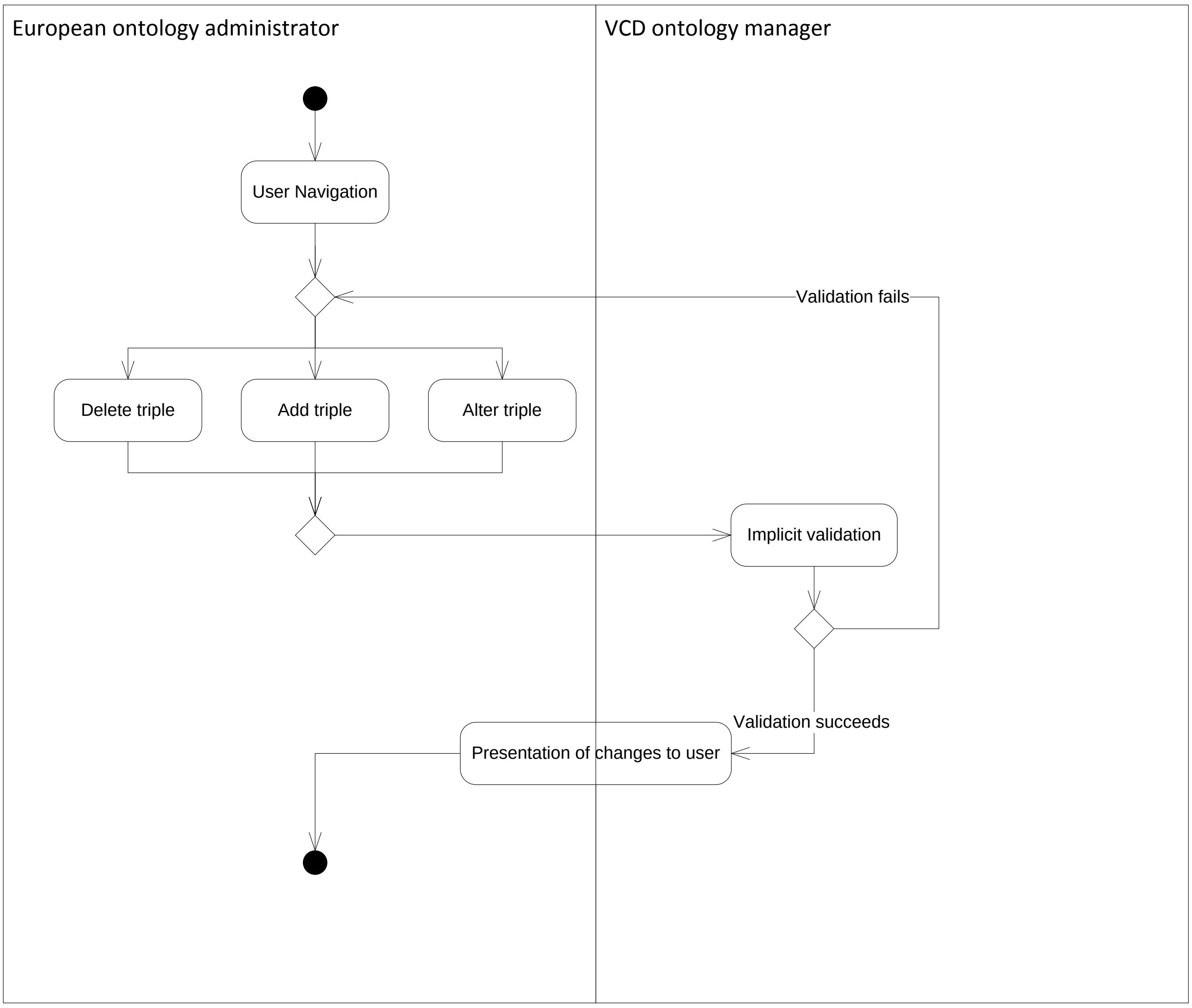


Figure 2-4: Activity diagram for the use case "Modify European ontology part

2.3 Use-Case group “Publication of national subsets of the rule set to the European VCD System”

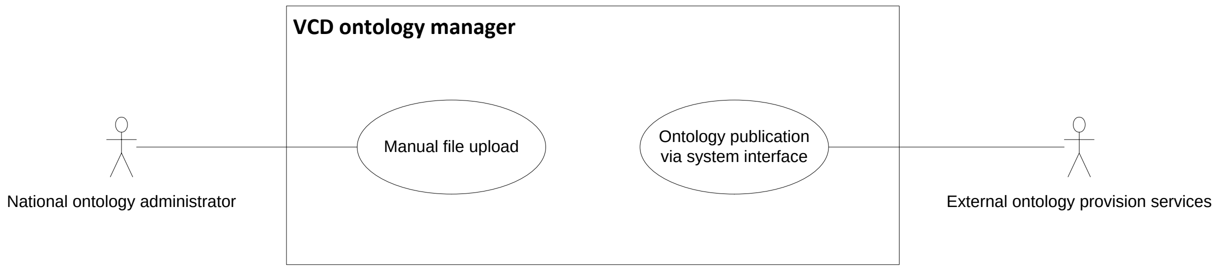

VCD ontology manager
Manual file upload
Ontology publication via system interface
National ontology administrator
External ontology provision services

Figure 2-5: Use-Cases of the VCD Ontology Maintenance Tool concerning publication of national subsets of the rule set to the European VCD System

Aspect	Description
Objective	An authorized user wants to upload national parts of the ontology to the VCD Ontology Maintenance Tool.
Results (postconditions in case of success)	User is logged into the system National part of the ontology is valid
Precondition	User is logged into the system National part of the ontology is valid
Postcondition in case of failure	User is logged into the system National part of the ontology is valid
Actor(s)	National ontology administrator
Initiating event	The user initiates the upload of the national subset of the ontology.
Description of interaction procedure with European VCD System (standard run)	The system provides a user interface which enables the user to upload a file containing the ontology parts. The user selects a local file containing the ontology parts. The system performs the compliance and access-right-checks for the uploaded parts of the ontology (with regard to 0 and 0). Positive compliance check: the system accepts the changes and performs the versioning defined in □. The system suggests the download of the updated (part of the) ontology using the functionality described in.
Description of interaction procedure with European VCD System (alternative runs)	Negative compliance check: the system performs the failure-steps defined in 0. The user can modify the uploaded ontology part directly in the VCD ontology management tool (Use-Case “Modify national ontology part”).
Extension(s)	None

Table 2-3: Manual file upload

The activity diagram in Figure 2-6 depicts the scenario described above.

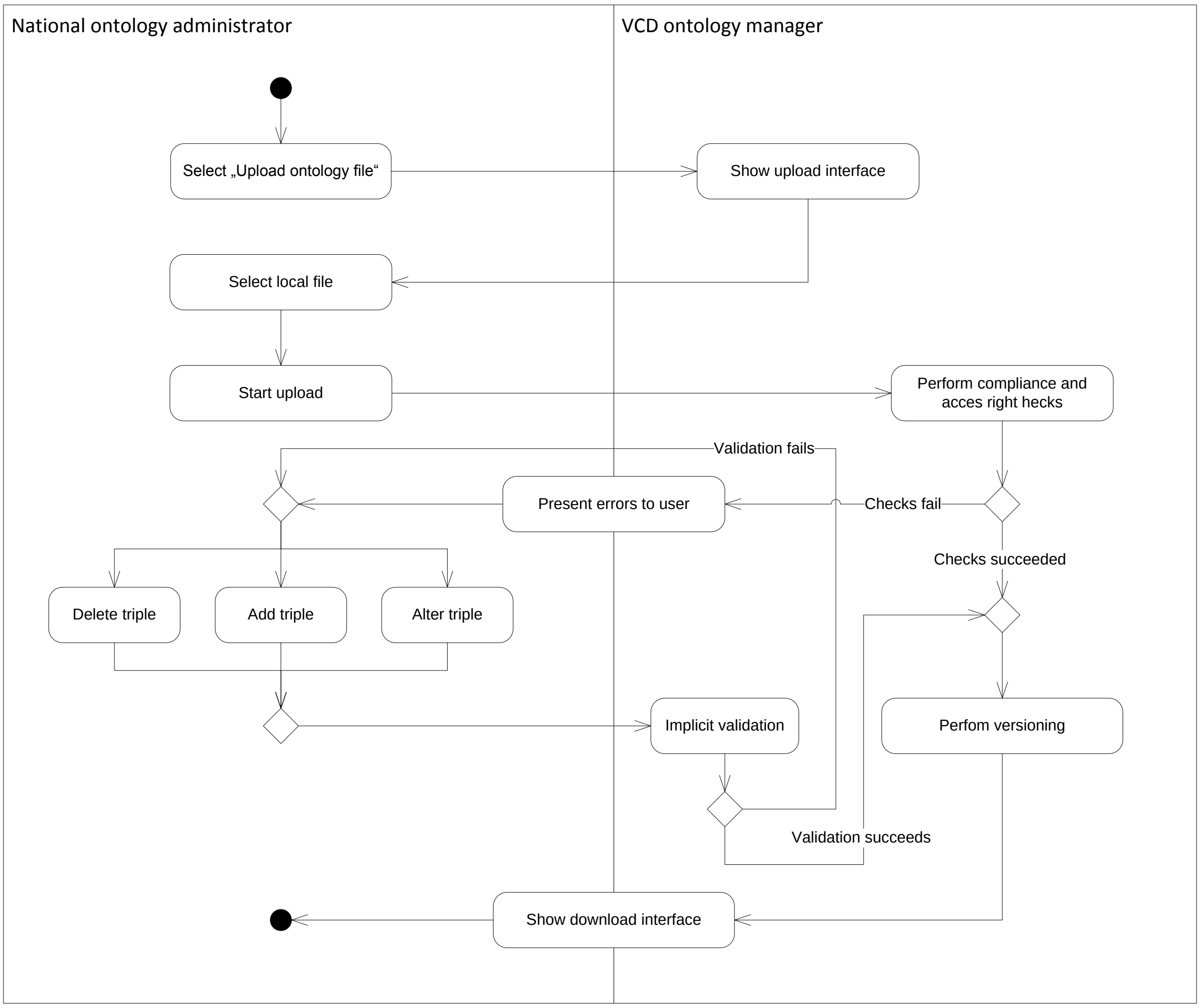


Figure 2-6: Activity diagram for the use case "Manual file upload"

Aspect	*Description*
Objective	A user in an external ontology provision system (e.g. as part of a VCD service) wants to upload national parts of the ontology to the VCD Ontology Maintenance Tool.
Results (postconditions in case of success)	National part of the ontology is valid
Precondition	External service has connectivity to the VCD Ontology Maintenance Tool National part of the ontology is valid
Postcondition in case of failure	National part of the ontology is valid
Actor(s)	External ontology provision system
Initiating event	The user initiates the upload of the national subset of the ontology via an external service.
Description of interaction procedure with European VCD System (standard run)	The system receives the uploaded ontology part The system performs the compliance and access-right-checks for the uploaded parts of the ontology (with regard to 3.1.1.2 and 3.1.1.3). Positive compliance check: the system accepts the changes and performs the versioning defined in □.
Description of interaction procedure with European VCD System (alternative runs)	Negative compliance check: the system performs the failure-steps defined in 0.

Extension(s)	None

Table 2-4: Ontology publication via system interface

The activity diagram in Figure 2-7 depicts the scenario described above.

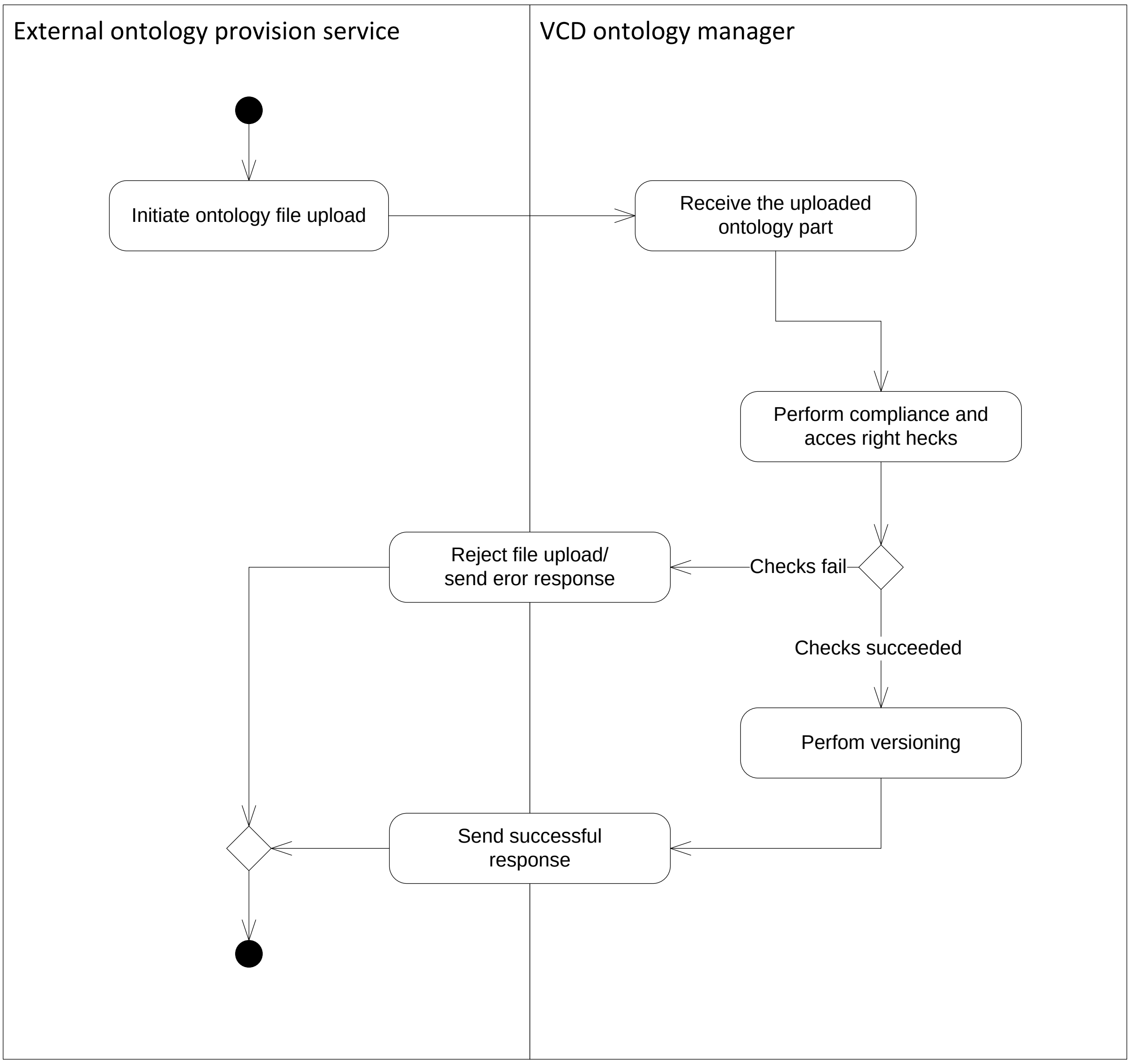

External ontology provision service
VCD ontology manager
Initiate ontology file upload
Receive the uploaded ontology part
Perform compliance and acces right hecks
Reject file upload/ send eror response
Checks fail
Checks succeeded
Perfom versioning
Send successful response

Figure 2-7: Activity diagram for the use case "Ontology publication via system interface

2.4 Use-Case group “Notification service”

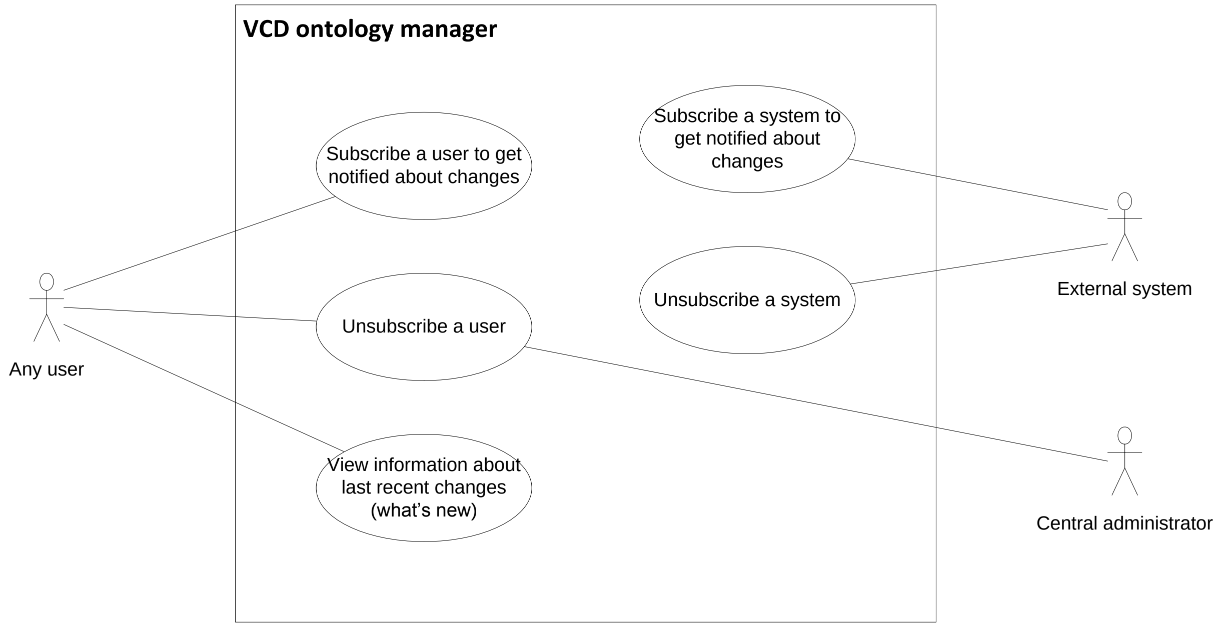

VCD ontology manager
Subscribe a user to get notified about changes
Subscribe a system to get notified about changes
Unsubscribe a user
Unsubscribe a system
View information about last recent changes (what’s new)
Any user
External system
Central administrator

Figure 2-8: Use-Cases of the VCD Ontology Maintenance Tool concerning notification service

Aspect	Description
Objective	A user wants to be informed about changes in the ontology.
Results (postconditions in case of success)	The user is subscribed for the notification about changes.
Precondition	
Postcondition in case of failure	The user is not subscribed for the notification about changes.
Actor(s)	User of the European VCD System
Initiating event	The user initiates the subscription functionality in the system.
Description of interaction procedure with European VCD System (standard run)	The system presents a form to the user where he can choose the mode of notification (if more than one exists) and defines the query (pattern) (e.g. parts of the ontology, ontology subset …) by either providing a new one or choosing an existing one. The system stores the new subscription.
Description of interaction procedure with European VCD System (alternative runs)	None
Extension(s)	None

Table 2-5: Subscribe a user to get notified about changes

The activity diagram in Figure 2-9 depicts the scenario described above.

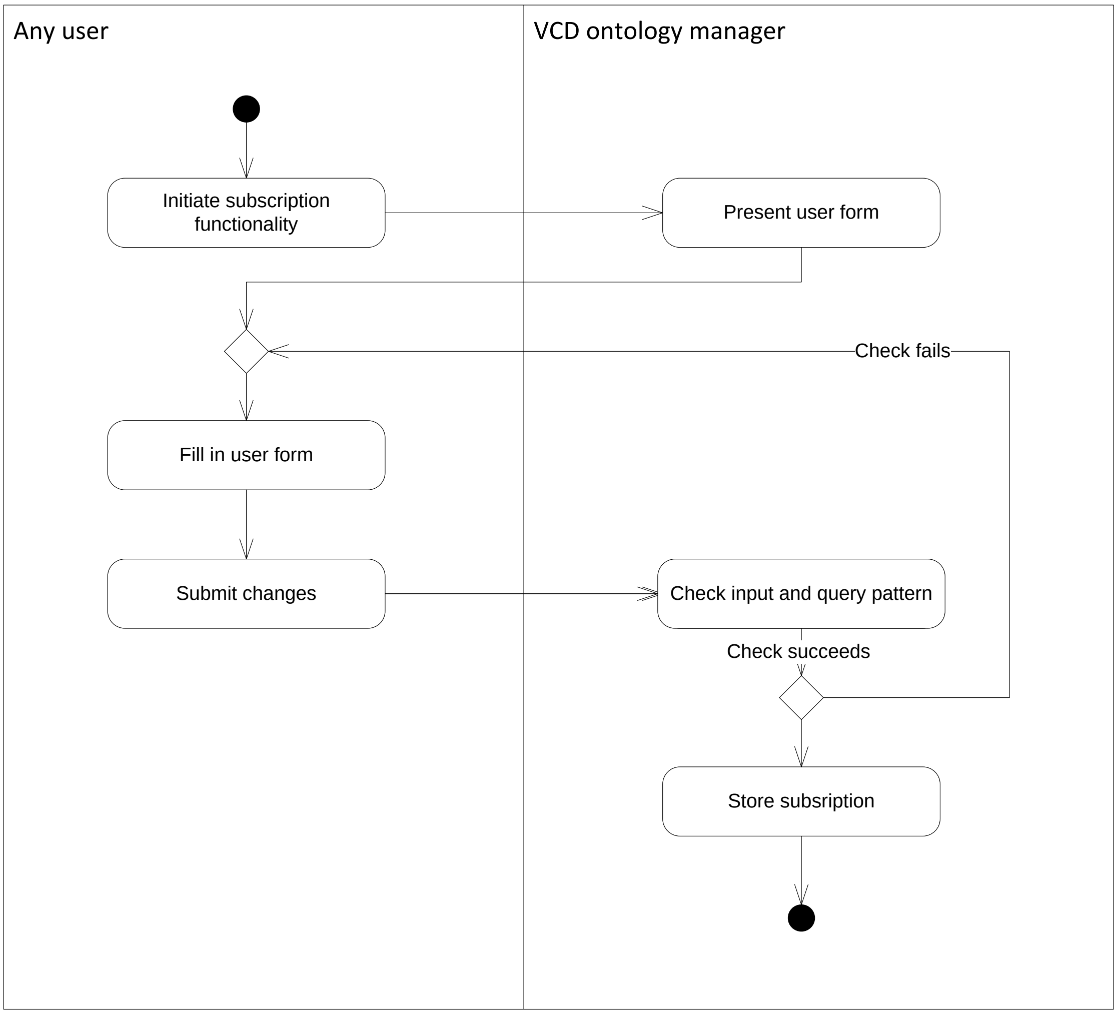


Figure 2-9: Activity diagram for the use case "Subscribe a user to get notified about changes"

Aspect	Description
Objective	A system needs to be notified about changes in the ontology.
Results (postconditions in case of success)	The system is subscribed for the notification about changes.
Precondition	
Postcondition in case of failure	The system is not subscribed for the notification about changes.
Actor(s)	An external system
Initiating event	The user initiates the subscription functionality in the external system.
Description of interaction procedure with European VCD System (standard run)	An external service calls the “Subscribe system-notification”-service provided by the VCD Ontology Maintenance Tool having the mode of notification (if more than one exists) and the query (pattern) (e.g. parts of the ontology, ontology subset …) provided as an input parameter. The system stores the new subscription.
Description of interaction procedure with European VCD System (alternative runs)	None
Extension(s)	None

Table 2-6: Subscribe a system to get notified about changes

The activity diagram in Figure 2-10 depicts the scenario described above.

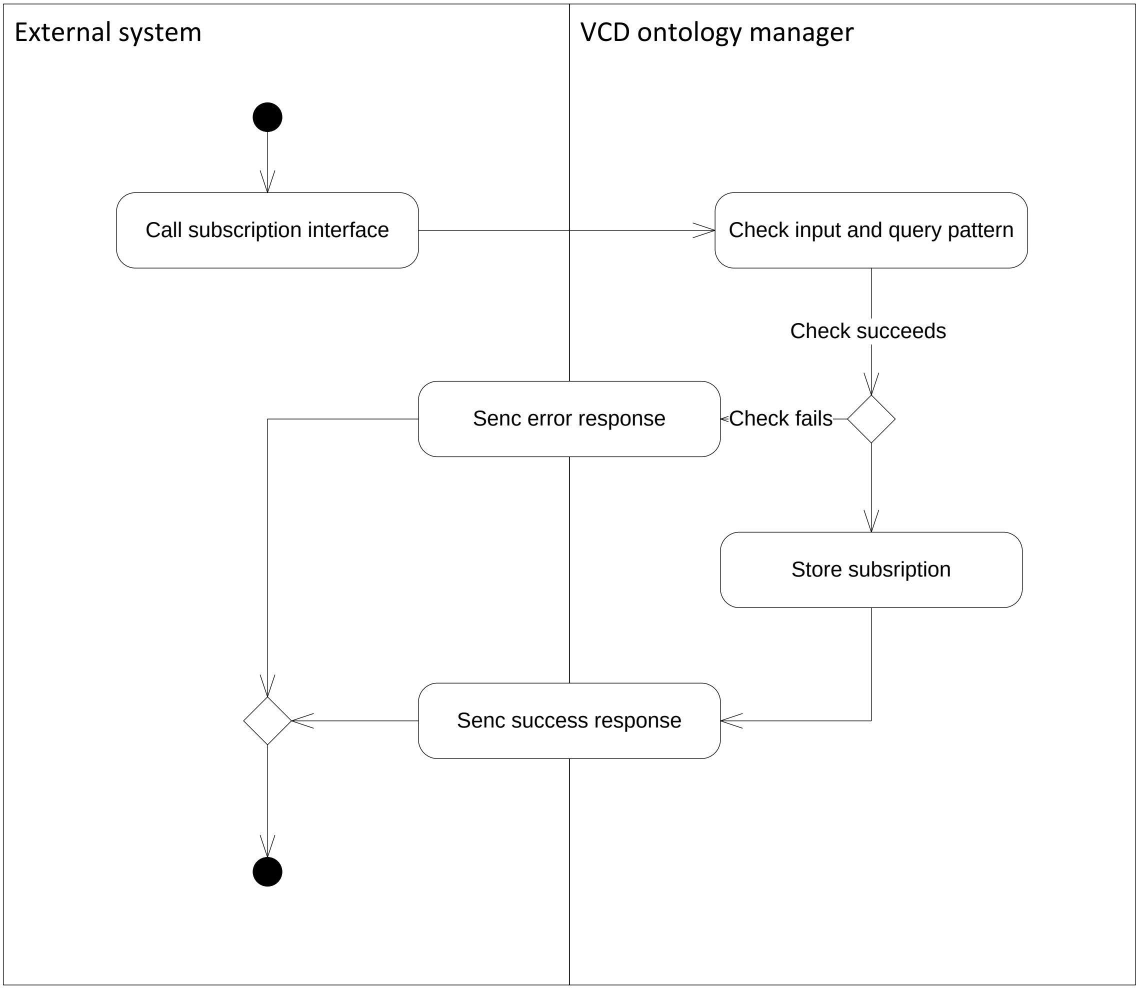


Figure 2-10: Activity diagram for the use case "Subscribe a system to get notified about changes"

Aspect	Description
Objective	A user no longer wants to be informed about changes in the ontology.
Results (postconditions in case of success)	The user is unsubscribed for the notification about changes.
Precondition	The user is subscribed for the notification about changes.
Postcondition in case of failure	The user is not unsubscribed for the notification about changes.
Actor(s)	User of the European VCD System A central user administrator acting for a user
Initiating event	The user initiates the unsubscription functionality.
Description of interaction procedure with European VCD System (standard run)	The user chooses the subscription he wants to unsubscribe to out of the list of his existing subscriptions. The system deletes the subscription.
Description of interaction procedure with European VCD System (alternative runs)	None
Extension(s)	None

Table 2-7: Unsubscribe a user

The activity diagram in Figure 2-11 depicts the scenario described above.

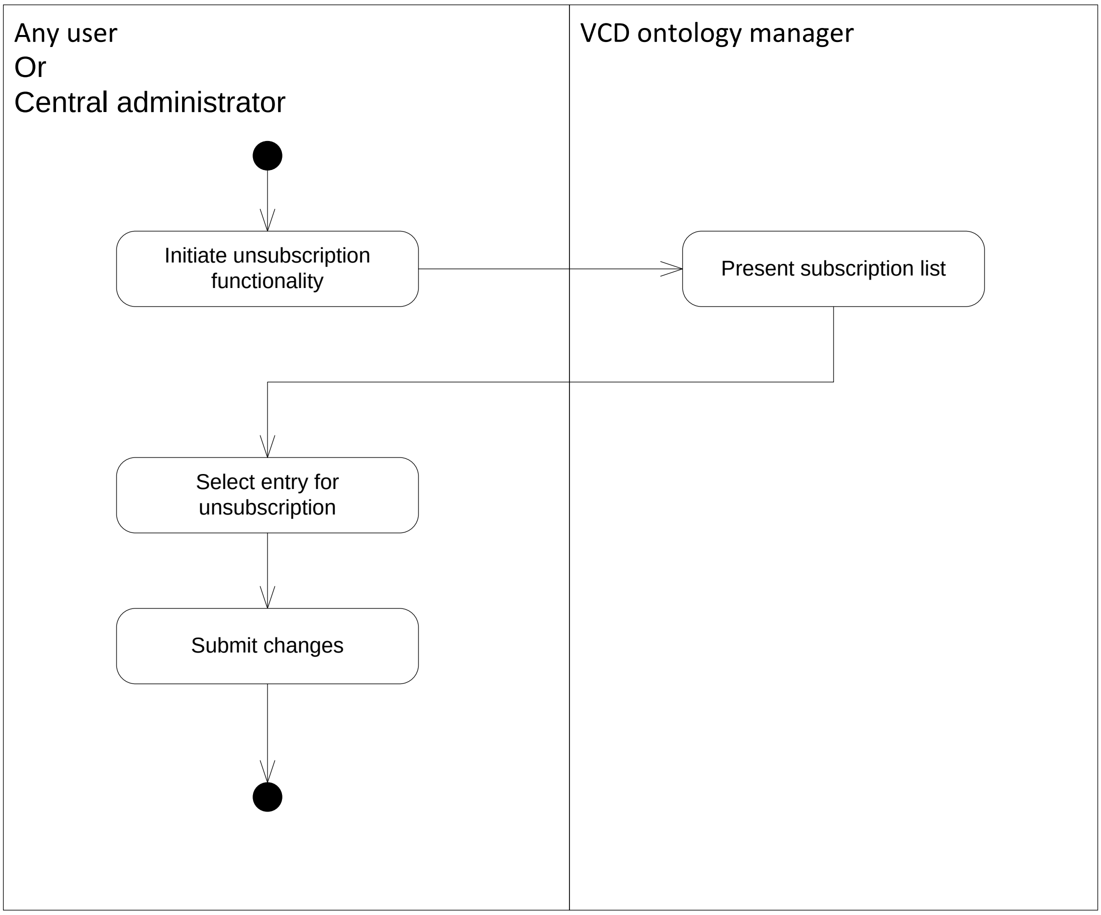

Any user
Or
Central administrator
VCD ontology manager
Initiate unsubscription functionality
Present subscription list
Select entry for unsubscription
Submit changes

Figure 2-11: Activity diagram for the use case "Unsubscribe a user"

Aspect	Description
Objective	A system no longer needs to be notified about changes in the ontology.
Results (postconditions in case of success)	The system is unsubscribed for the notification about changes.
Precondition	The system is subscribed for the notification about changes.
Postcondition in case of failure	The system is not unsubscribed for the notification about changes.
Actor(s)	An external system
Initiating event	The user initiates the unsubscription functionality in the external system.
Description of interaction procedure with European VCD System (standard run)	The external services calls the “Unsubscribe system-notification”-service provided by the VCD Ontology Maintenance Tool, having the subscription which should be unsubscribed provided as an input parameter. The system deletes the subscription.
Description of interaction procedure with European VCD System (alternative runs)	None
Extension(s)	None

Table 2-8: Unsubscribe a system

The activity diagram in Figure 2-12 depicts the scenario described above.

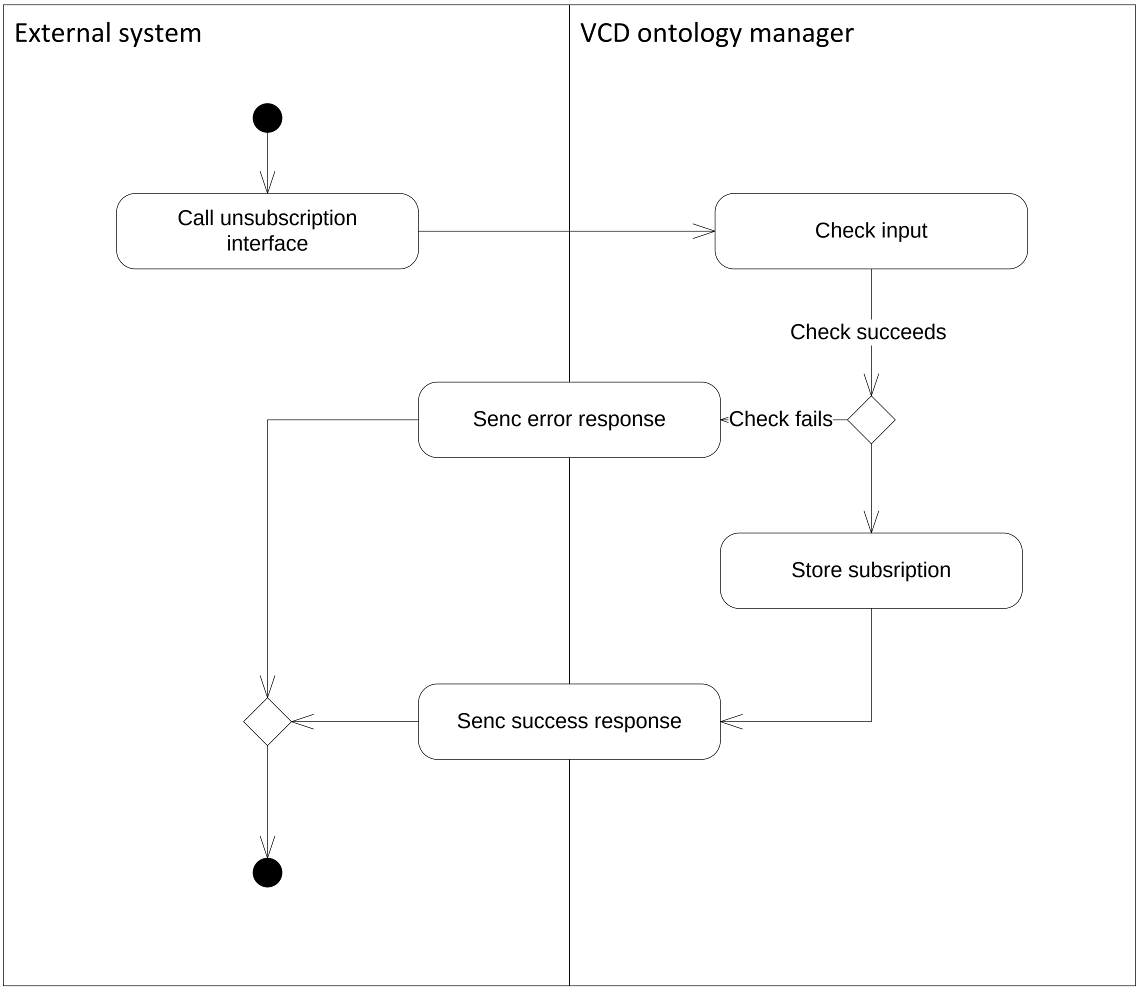


Figure 2-12: Activity diagram for the use case "Unsubscribe a system"

Aspect	Description
Objective	A user wants to view the last recent changes in the ontology.
Results (postconditions in case of success)	The last recent changes in the ontology are displayed in a list.
Precondition	
Postcondition in case of failure	
Actor(s)	User of the European VCD SystemUser of the European VCD System
Initiating event	The user initiates the “view last recent changes” functionality.
Description of interaction procedure with European VCD System (standard run)	The user chooses one of the provided sets of “What’s new histories” he wants to view (whole ontology, upper level ontology, European ontology, any national ontology). The system presents a list of the last recent changes in the selected (sub-) ontology.
Description of interaction procedure with European VCD System (alternative runs)	None
Extension(s)	None

Table 2-9: Unsubscribe a system

The activity diagram in Figure 2-13 depicts the scenario described above.

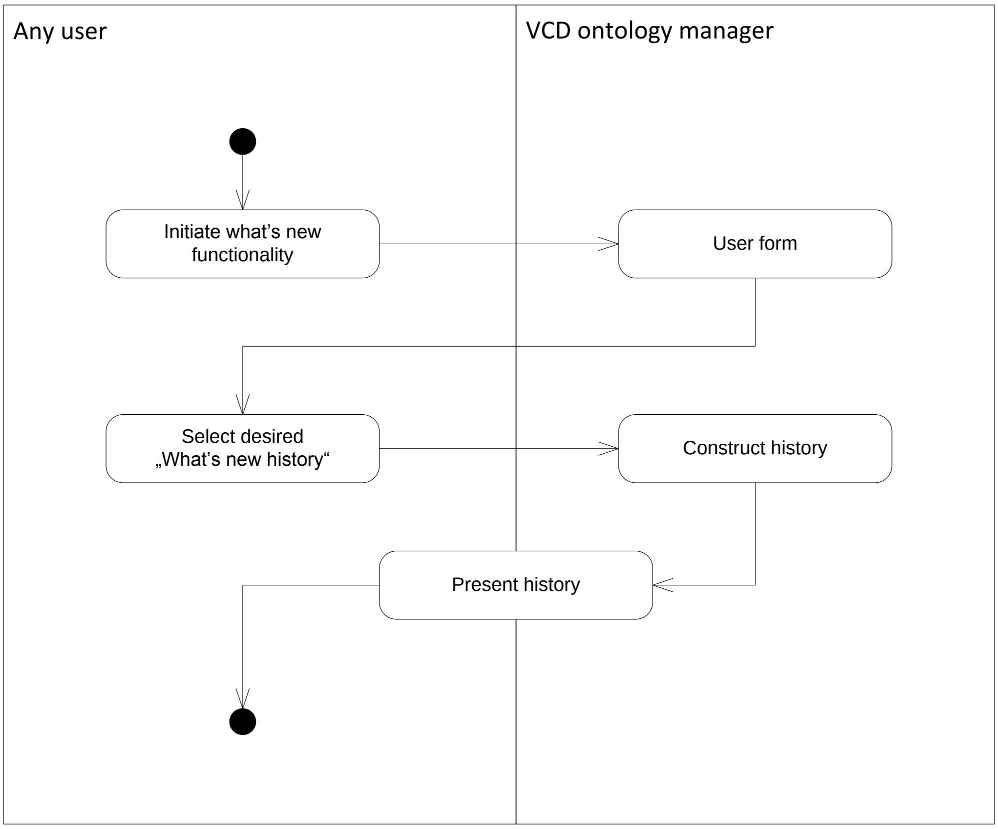


Figure 2-13: Activity diagram for the use case "View information about last recent changes (what's new)"

2.5 Use-Case group "System administration"

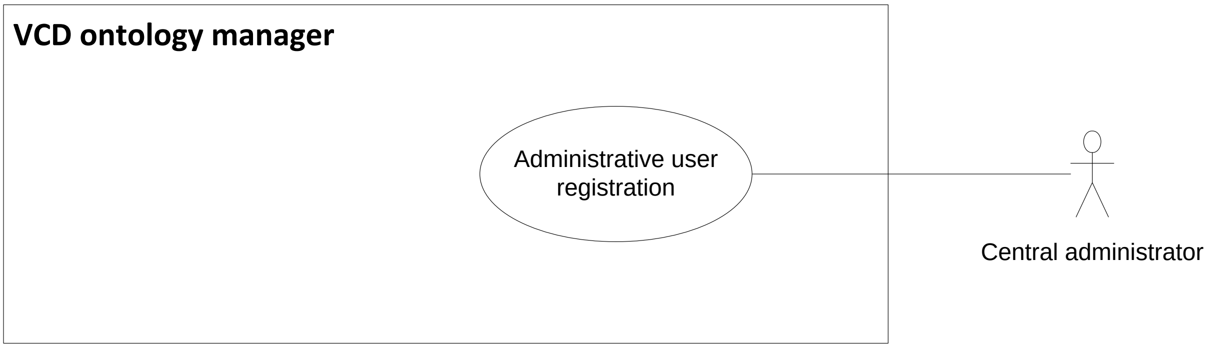


Figure 2-14: Use-Cases of the VCD Ontology Maintenance Tool concerning system administration

Aspect	Description
Objective	The registration information (incl. access rights) of an administrator (National ontology administrator, European ontology administrator, Central administrator) of the VCD Ontology Maintenance Tool should be modified.
Results (postconditions in case of success)	The registration information (incl. access rights) of the administrator is modified.
Precondition	At least one central administrative user account exists.
Postcondition in case of failure	The registration information (incl. access rights) of the administrator is not modified.
Actor(s)	Central user administrator
Initiating event	The user logs in as a central administrator.
Description of interaction procedure with European VCD System (standard run)	The user selects an existing user account and opens it for modification The user edits the role of the user account. The roles are defined in □. The user stores the user account The system notifies the user which is represented in the modified account about the changes
Description of interaction procedure with European VCD System (alternative runs)	None
Extension(s)	None

Table 2-10: Administrative user registration

The activity diagram in Figure 2-15 depicts the scenario described above.

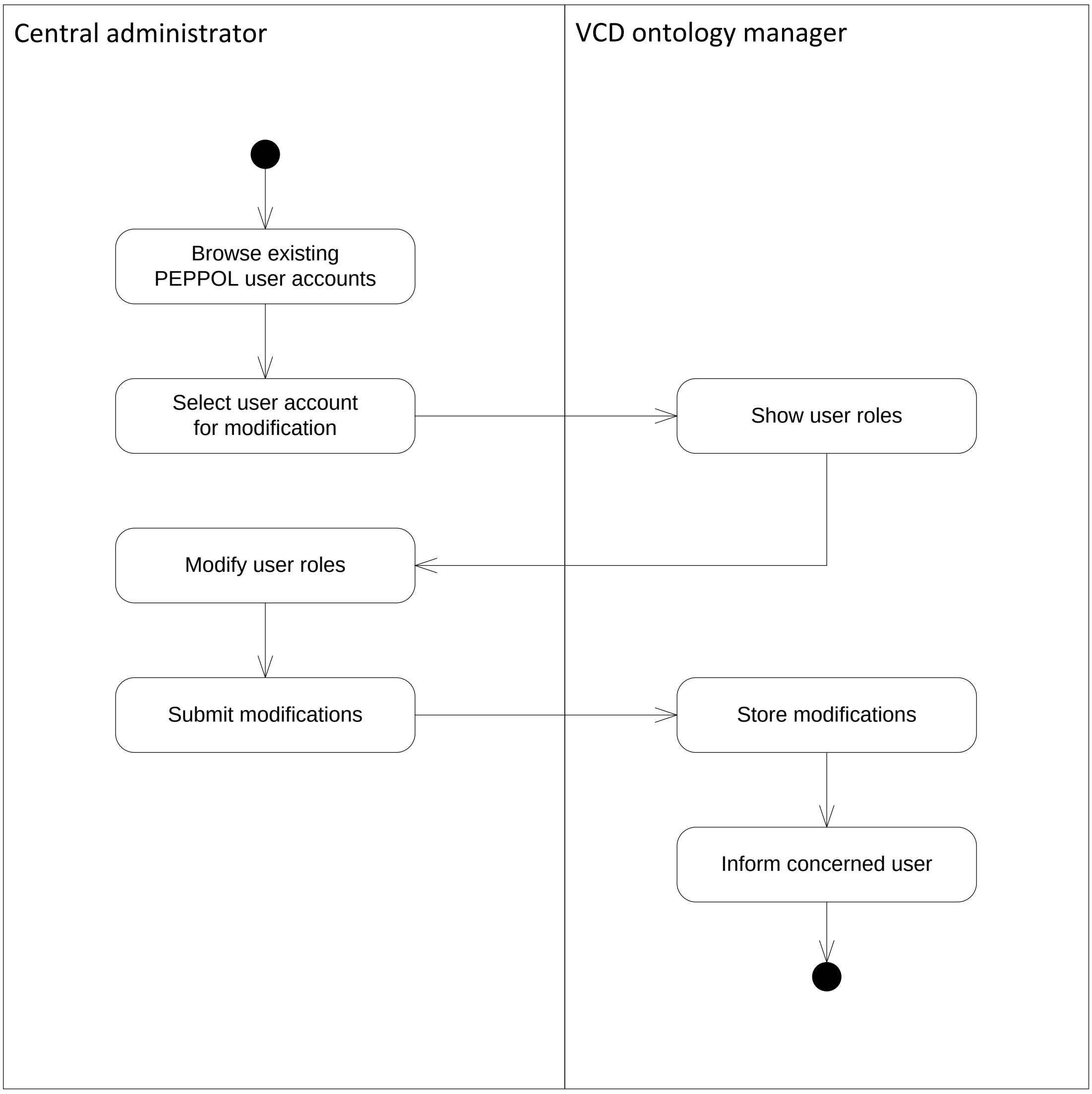


Figure 2-15: Activity diagram for the use case "Administrative user registration"

3 General system functionality

- Logging of changes

The system logs the following information for predefined types of modification to the ontology:

- User
- Date/time
- Description of the change (change set) [optional]

The predefined types of modification have to be configured in the system.

- Transaction security

The system provides ACID security (e.g. locking parts of the ontology/triples or subsets of the ontology when working with it).

3.1 User registration, system registration and access management

- User Roles

The system supports a user role concept with the following capabilities.
A user role consists of the following information:

- Access rights
- Read
- Modify (modify, read and delete)
- Lock ontology part/subset
- Query pattern (defining those parts of the ontology, the access right is applicable to)

Additionally, the following administrative user roles are required:

- Central administrator
- European ontology administrator
- National ontology administrator (for each country)

- Access management and user registration

The system provides an access management component which is able to store, retrieve and modify the following information:
System users (user name, password, company information tbd…)

Roles and access rights for the system users

- Registry for external ontology provision services and systems

The system provides a management component which is able to store, retrieve and modify the following information:
External ontology provision services (Service name, service provider information tbd, service access parameters…)

Access rights for services (service can act similar to a National ontology administrator)

3.2 Core functions

- Multilinguality

The user interface is available in the official languages of the European Union.

- Ontology browsing interface

The system must be able to visualize the ontology, ontology parts and ontology subsets of it to the user. This visualization should be based on concepts like class diagrams, lists and tables or other more sophisticated visualization concepts.
The system should be able to group parts of the ontology and to provide a zoom in/out functionality.
The system should provide an automated layouting mechanism.

- Ontology modification interface (Part of Ontology Manager)

The system must be able to visually edit the ontology, ontology parts and ontology subsets of it to the user. The system must visualize parts of the ontology which are editable by a user in a different way than the parts of the ontology which are not editable by the user.
The system must visualize the result of the validity check to the user, including the implications on other parts of the ontology.
The system validates for each visualized ontology part whether the user has the necessary access rights to view this part and to modify to this part. If the user does not have the “read”-access right, the ontology part is not displayed. If the user does not have the “modify” access right, the ontology part cannot be modified in the user interface.

- Manual ontology upload (Part of Ontology Manager)

The system must provide a user interface for uploading parts of the ontology (being pre-defined models, e.g. national subset, European subset, upper level ontology…). The file format for this upload must be defined (e.g.n3). If the upload fails, the system informs the user about the failure. After a successful upload has been completed, the system performs the function “Merge ontologies”. The system informs the user about the result of the merging activity. If the merging has resulted in errors, these errors are shown to the user.

- Ontology import system interface (Part of Ontology Manager)

The system must provide a system interface for importing parts of the ontology (being pre-defined models, e.g. national subset, European subset, upper level ontology…). The format and payload of this interface must be defined (e.g. wsdl). After a successful interface call has been completed, the system performs the function “Merge ontologies”. The system responds to the initial interface call. If the import or the merging fails, the system reacts with an error response.

- Ontology merging (Part of Ontology Manager)

The system must provide a functionality to merge new ontology parts into the existing ontology.
The subject of a merge can only be one or more pre-defined models. The process of merging consists of:

- Compliance check
- Deleting the old model entirely
- Adding the new model
- Triples outside the model, which refer to this model, are not affected.

- Ontology graph management (Part of Ontology Manager)

The system operates upon a virtual consolidated graph view. This graph combines all ontology parts potentially stored in different data sources (triple sources like n3-files or triple stores). The system must able to provide this virtual consolidated graph for the other system parts.

- Ontology querying (Part of Ontology Manager)

The system must provide an ontology query engine which support different querying mechanisms (e.g. sparqle).

The query engine must provide a user interface formulate and trigger the query and to visualize the query results (e.g. as a list).

- Notification

Notification subscription/unsubscription
The system must provide a functionality to store notification subscriptions. A notification subscription consists of the following elements:

- Subscribing user or system
- Mode of notification (e.g. E-Mail, RSS)
- Query (context of change)

The system must provide a user interface to create, modify and delete such subscriptions. The system should suggest predefined query (patterns) (e.g. parts of the ontology, ontology subset …), which can be chosen by the user when creating or modifying a subscription.

Notification execution

Active notification of a subscribed user about changes in the ontology
The system informs subscribed users about changes in the ontology, whenever a change occurs which matches the subscription pattern (e.g. E-Mail, RSS-Feed …). This notification just contains the information that there has been a change concerning the subscribed pattern and the pattern itself.

- Ontology versioning

Create Versioning information (Part of Ontology Manager)
The system must provide a functionality to create versioning information for parts of the ontology.
Whenever the ontology is modified (either by a manual process or by a publication process), the system must add versioning information for these modification in order to be able to view and to trace modifications ("change log"). The following versioning information has to be included:

- Version number
- Change date/time
- Changed by user
- Annotation (reason for change, e.g. legal reference)

Version history (Part of Ontology Manager)
The system must provide a functionality to keep trace of the version history of parts of the ontology. The system must provide a user interface to present the version history of specified ontology concepts (e.g. class, property…).

- Ontology export

The system must provide a functionality to export pre-defined ontology parts (models) in a specific ontology format (e.g. n3) and make accessible for either a user download functionality or system interface download functionality.

- Ontology validation

Ontology compliance check (Part of Ontology Manager)
The following checks are performed by the system:
Modifications must be conform to the schema (OWL DL compliant). E.g. every national evidence has to be linked to one or more national criteria (constraint)

Modifications must be conform to additional rules which are interpreted by the (extended) reasoner. E.g. is the European ontology administrator allowed to delete a European criterion, which has active mappings to national criteria, or may he just rename it

Failure steps after negative compliance check (Part of Ontology Manager)

In case of manual user modification or manual ontology upload, the system provides detailed information about violated ontology parts (e.g. URIs, label, violation problem (e.g. constraint violation)). In case of ontology publication via a system interface, the system rejects the publication.

Ontology access right check (Part of Ontology Manager)
The system validates for each automated modification whether the user or external system has the necessary access rights to perform the modification. If the access rights of the user or external system are not sufficient, the system reacts with an error.

Report about relevant affected concepts due to changes in the ontology
The system generates a list of affected concepts resulting from types of modification by the user (e.g. if the user deletes an upper level class, he should be informed about the affected properties, subclasses and instances of this class). This list is presented to the user after he has performed the modification but before the modification is persisted. The user must have the possibility to accept or to reject the modification.

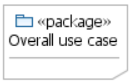
What's new output generation

Information about last recent changes (what's new) for users: The system provides a list of the last recent changes which have been made, in the whole ontology as well as in the upper level ontology, the EU-ontology and each national ontology.

- **Functional specification of VCD Ontology Interaction Tool**

The component *VCD Ontology Interaction Tool* of the European VCD System depends on the VCD Ontology which stores relevant mapping information.

3.3 Function ”Derive evidences”



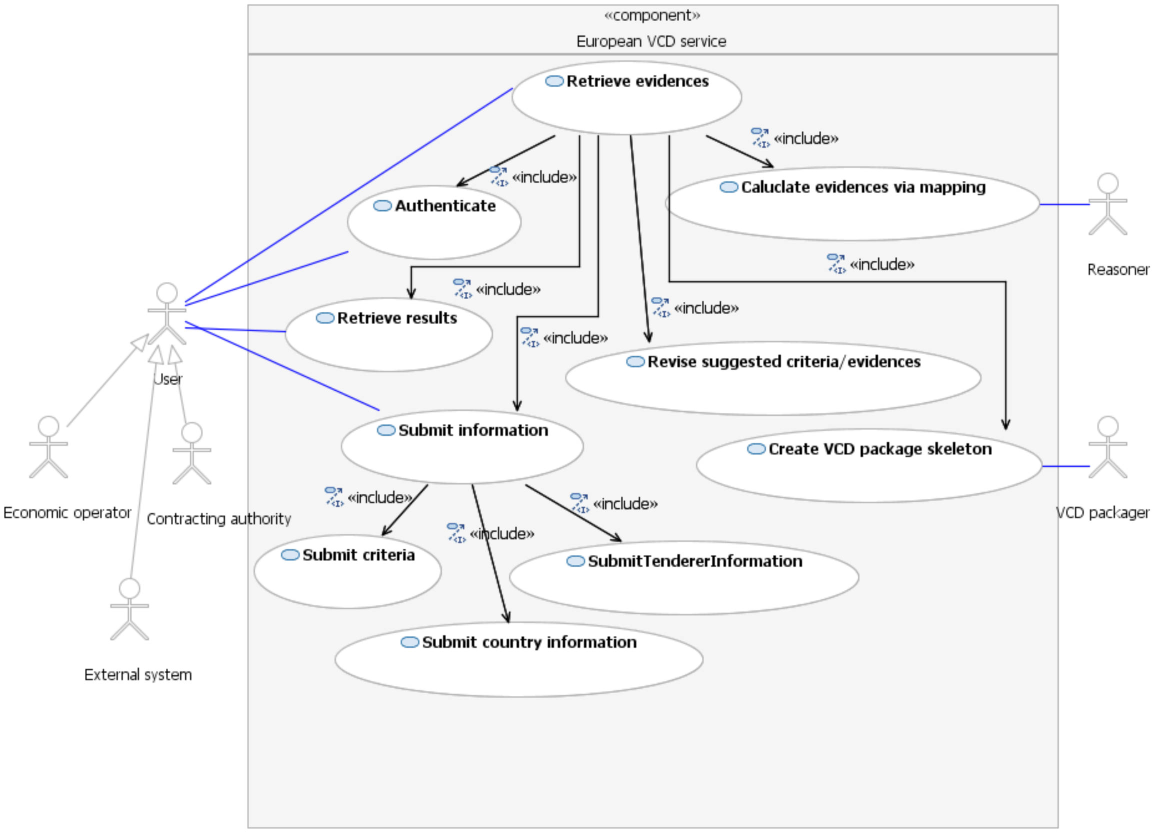


Figure 3-1: Functionalities of the European VCD Ontology Interaction Tool

Aspect	Description
Objective	Query the European VCD Ontology Interaction Tool to retrieve the list of evidences an economic operator from country A may provide that confer to a list of criteria set out in a call for tenders or contract notice of a contracting authority in country B respectively stated in the national legislation. The query should result in visualizing the mapping and in providing a list of evidences (e.g. via a VCD skeleton Container).
Precondition	The qualification criteria for each tenderer structure element for a specific Call for Tender (e.g. from a published contract notice) are known and available for input in order to be able to confirm or alter the criteria suggested by the system
Results (postconditions in case of success)	VCD skeleton container containing mappings of criteria to evidences
Postcondition in case of failure	The user gets informed about the reasons for failure (e.g. missing information).
Actor(s)	Economic operator Contracting authority User of the European VCD System
Initiating event	The user requests a list of evidences suitable to prove certain criteria according to the rule set (requests a VCD Skeleton Container)
Description of interaction procedure with European VCD System	The nationality of the contracting authority (e.g. from the CFT) and the necessary tenderer information (including tenderer constellation, legal form of tenderer(s), nationalities of tenderer(s)) for each tenderer structure element (Economic Operator) are entered into the system. The following possibilities exist: a) Manual data input by the user: The user enters the data into the European VCD System using an appropriate user interface. The interface is directly based on the Ontology, giving immediate support the user (e.g. by validating user input on-the-fly). b) Data input via system interface: The data is provided by another system (National VCD Systems) via a system interface in a structured way (VCD Pre-Skeleton Container) c) The user uploads an existing VCD container or VCD skeleton container. The system extracts the existing data out of the VCD (skeleton) container /. The user can modify the extracted data in order to create the necessary input. The system validates the input and requests corrections / completions if necessary. The system infers criteria for each Economic Operator within the the tenderer structure and suggests it to the user. The user confirms or alters the suggested criteria. The system infers a list of suitable evidences for each selected criteria and presents this information to the user (“mapping tree”). The system produces a VCD skeleton package and passes it to the requester (download for the user or via system interface to the National VCD System).The user can exit the system at any time.
Extension(s)	1a. Automatic retrieval of criteria information from TED. This functionality will be implemented in a post-PEPPOL scenario.-

Table 3-1: Use case description

The activity diagram in Figure 3-2 depicts the scenario described above.

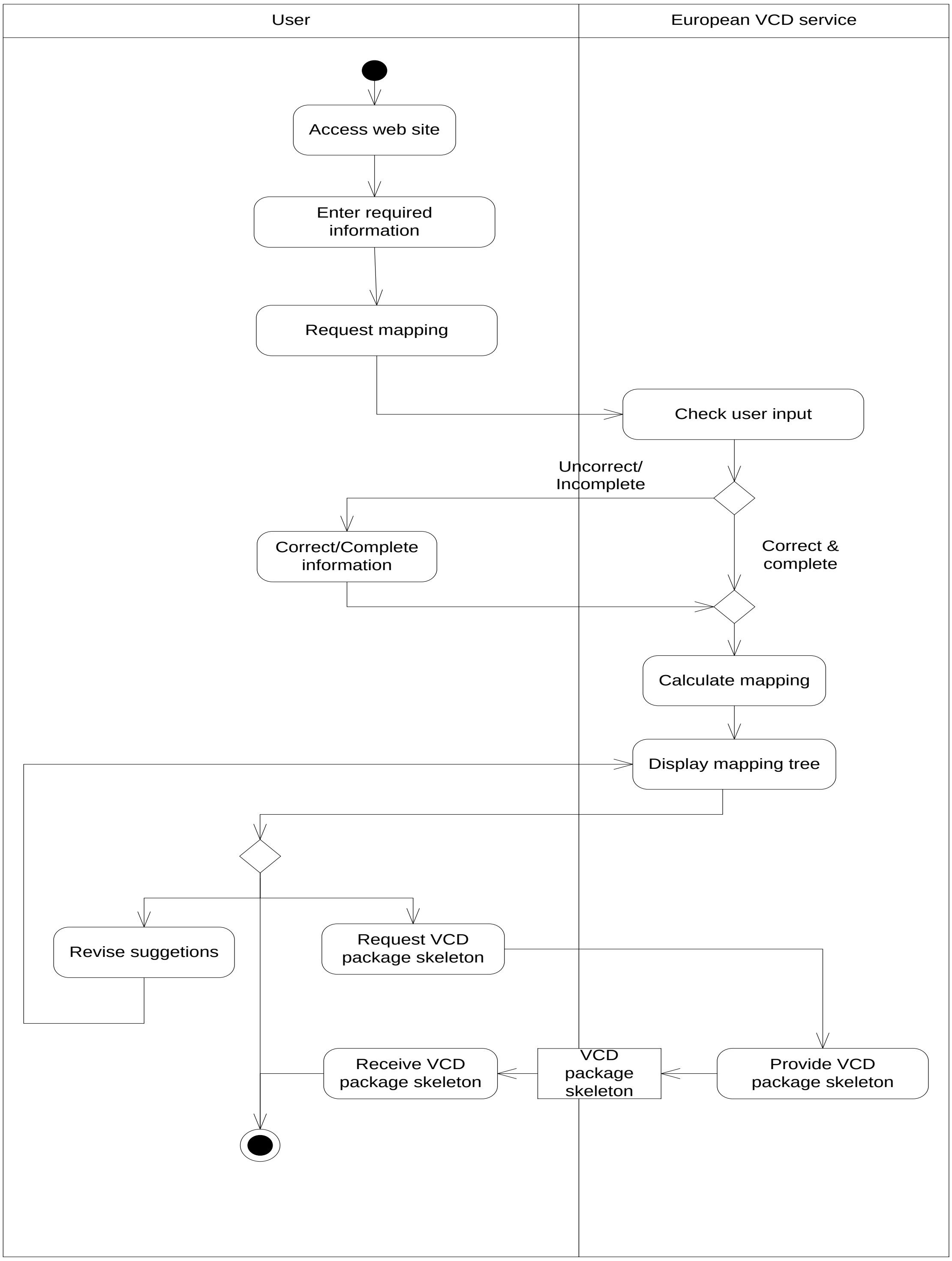


Figure 3-2: Activity diagram for the use case "Retrieve evidences"

- Function “Extraction of tenderer structure from existing VCD Skeleton Container”

As described above, besides the functionalities of manual data input by a user or data input via a system interface, the system should offer thge possibility to upload and import an existing VCD (Skeleton) Container. This functionality is described in this section.

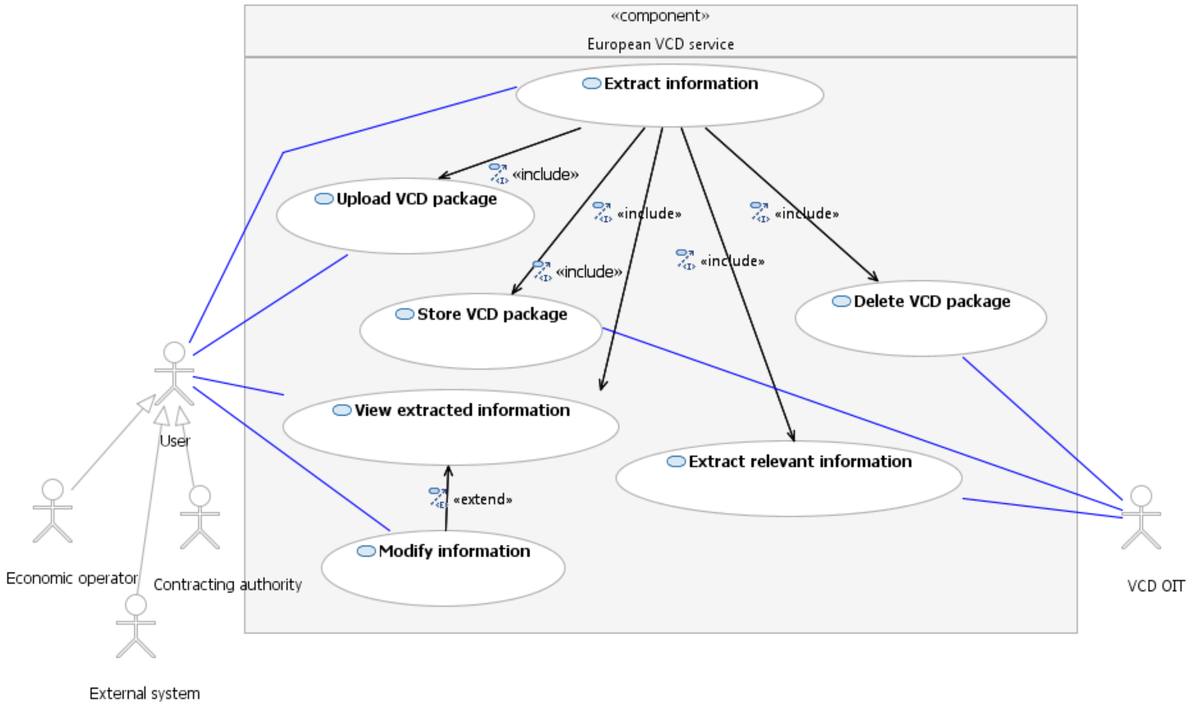


Figure 3-3: Usecase diagram for function “Extraction of tenderer structure from existing VCD (Skeleton) Container”

Aspect	Description
Objective	Extract relevant tenderer information in order to generate a VCD Skeleton Package and/or view appropriate Evidences from a previously created VCD (Skeleton) Container, omitting the need for a manual input of this data.
Precondition	The tenderer possesses at least one previously created VCD (Skeleton) Container. The European VCD System thereby accepts both, a VCD Container or just a VCD Skeleton Container.
Results (postconditions in case of success)	The European VCD System displays the extracted tenderer information.
Postcondition in case of failure	The user gets informed about the reasons for failure (e.g. missing information).
Actor(s)	Economic operator
Initiating event	The user requests the upload of a previously created VCD (Skeleton) Container for the extraction of the tenderer information.
Description of interaction procedure with European VCD System	The user selects the option ”Upload existing VCD” The user selects the path to the VCD (Skeleton) Cpontainer and starts the upload. The system stores the VCD (Skeleton) Container in a temporary folder. The system extracts the relevant tenderer information and displays these via the user interface and is allowing the user to modify the results. The system deletes the temporarily stored VCD (Skeleton) Container.
Extension(s)	-

Table 3-2: Use case description

The activity diagram in Figure 3-4 depicts the scenario described above.

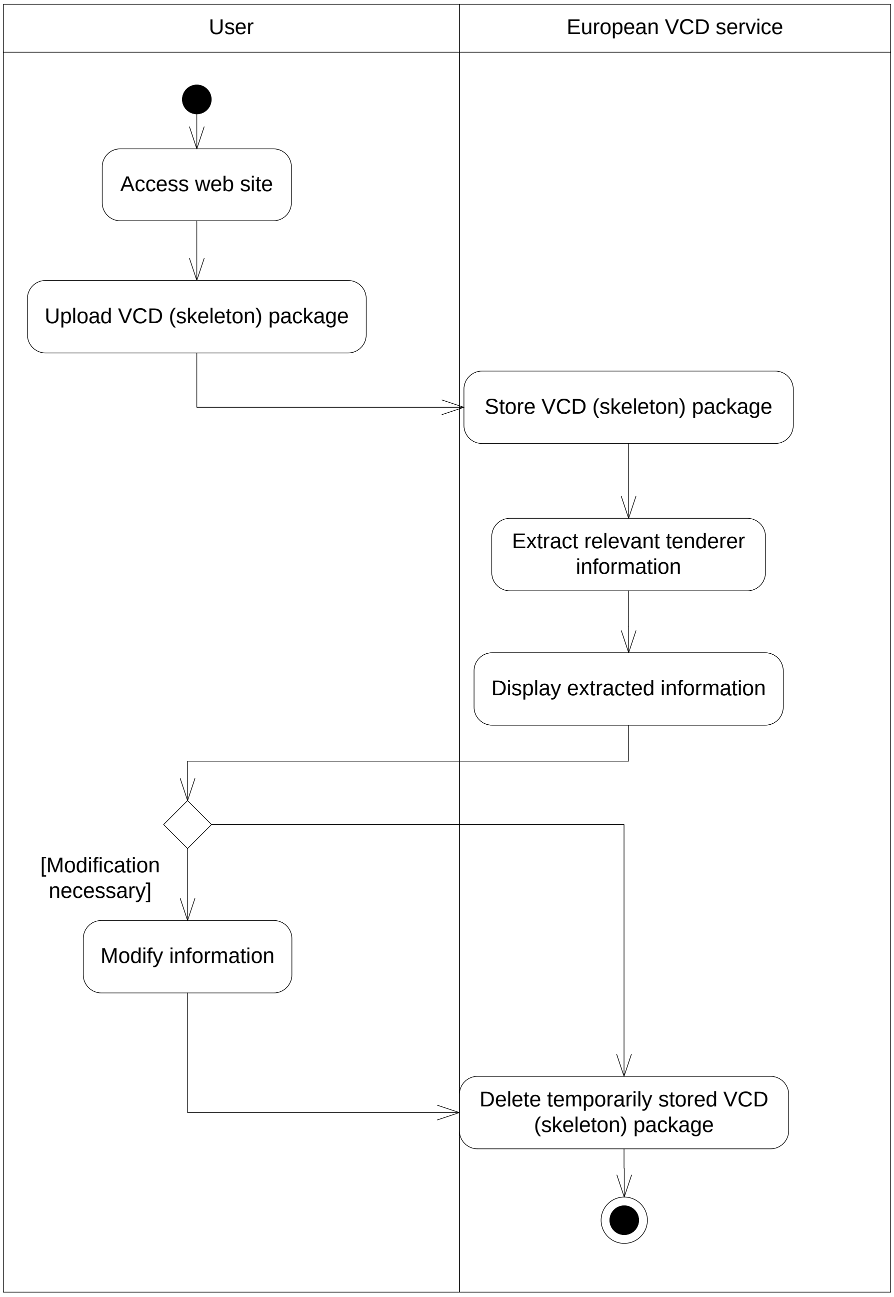

User
European VCD service
Access web site
Upload VCD (skeleton) package
Store VCD (skeleton) package
Extract relevant tenderer information
Display extracted information
[Modification necessary]
Modify information
Delete temporarily stored VCD (skeleton) package

Figure 3-4: Activity diagram for the use case "Extraction of tenderer structure from existing VCD (Skeleton) Container"

3.4 Function “Retrieve data”

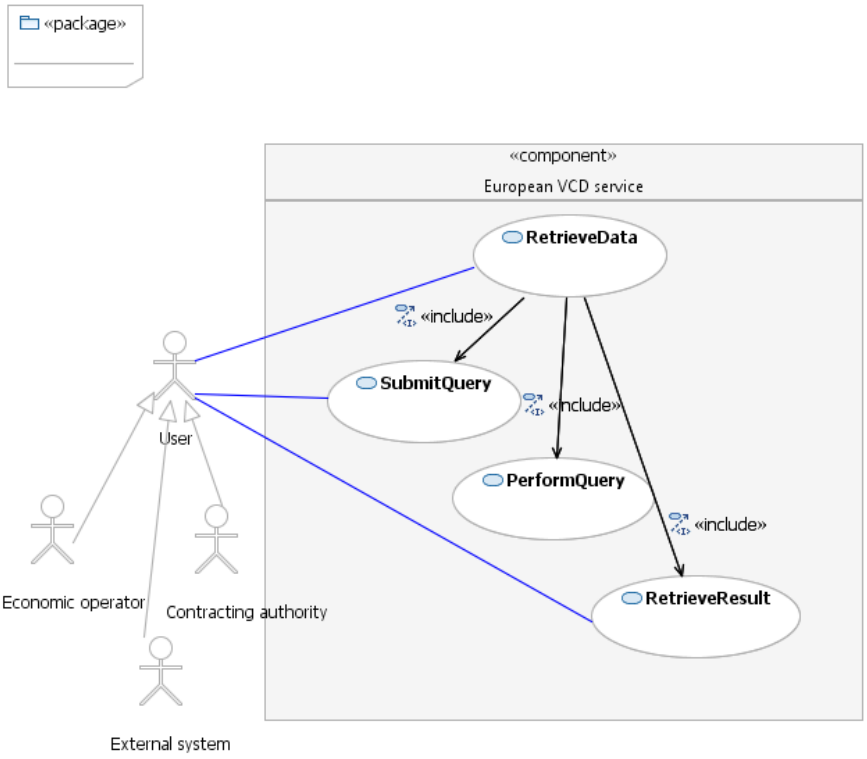


Figure 3-5: Querying data stored in the ontology

Aspect	Description
Objective	Query the European VCD System to retrieve any information/data stored in the ontology, e.g. information about all evidences available in a country.
Precondition	---
Results (postconditions in case of success)	Query results
Postcondition in case of failure	The user gets informed about the reasons for failure (e.g. missing information).
Actor(s)	Economic operator Contracting authority User of the European VCD System External system (Nb: also called “user” in the following)
Initiating event	The user requests the retrieval of data from the ontology (selecting menu item or link in European VCD Ontology Interaction Tool).
Description of interaction procedure with European VCD System	The user enters a query (e.g. SPARQL, Query pattern). The user requests the execution of the query. The system checks the user input for completion/errors and requests corrections/completions if necessary. The system uses the user input to search the ontology for the requested data. The system provides the results in downloadable form (format tbd). The user downloads the results.
Extension(s)	--

Table 3-3: Use case description

The activity diagram in Figure 3-6 depicts the scenario described above.

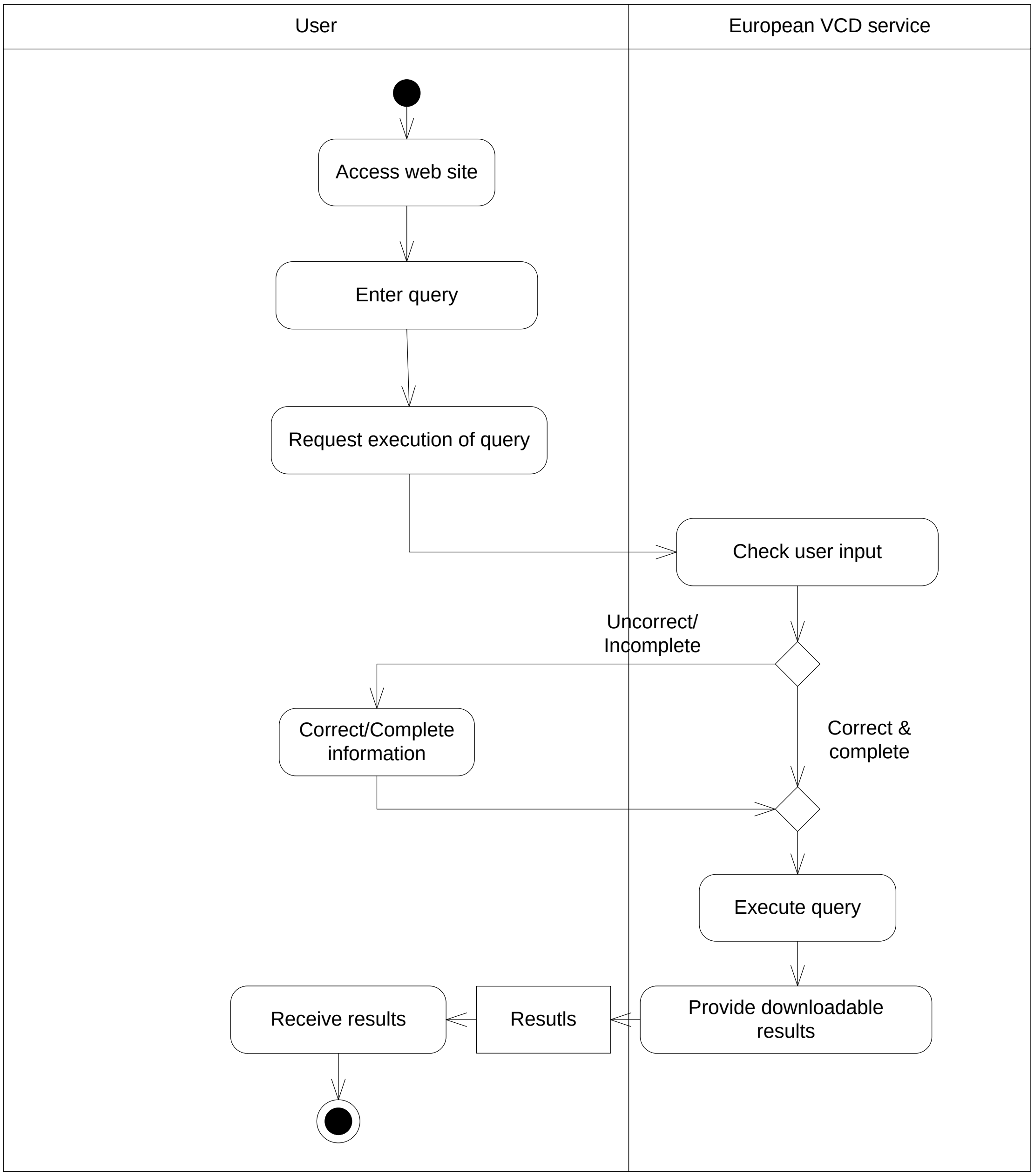


Figure 3-6: Activity diagram for the use case "Retrieve data"

3.5 Function “Download ontology”

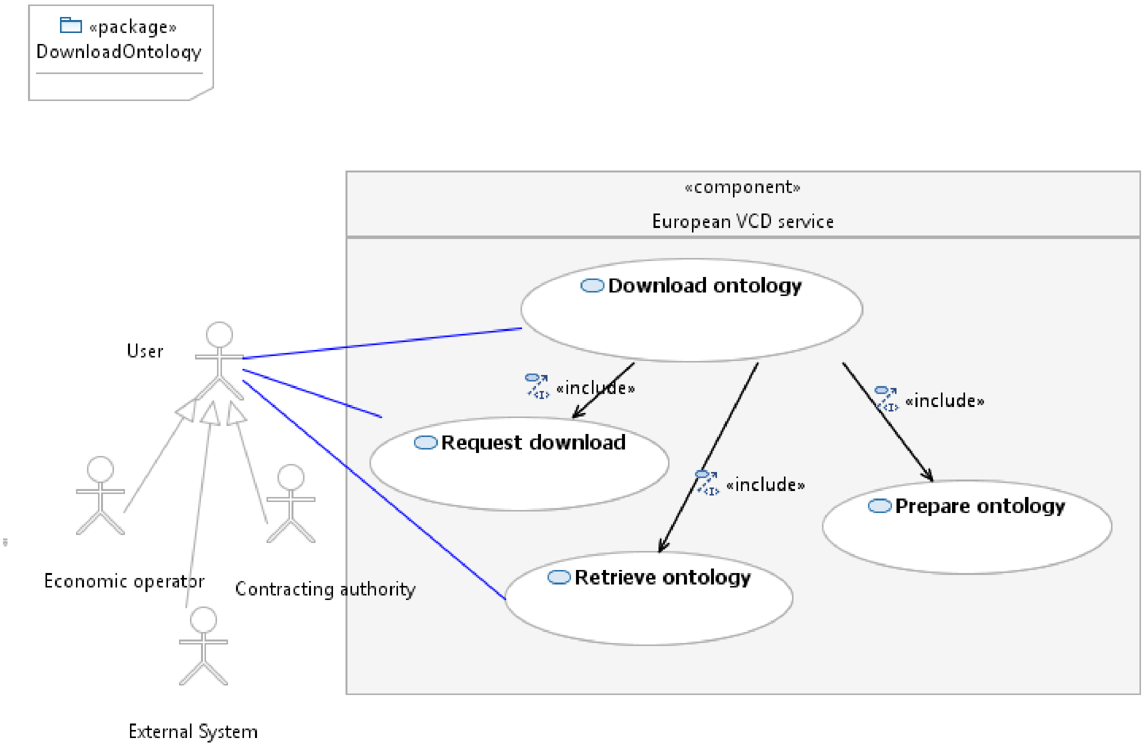


Figure 3-7: Querying for data stored in the ontology

Aspect	Description
Objective	Download the ontology.
Precondition	---
Results (postconditions in case of success)	The complete ontology in a defined downloadable format.
Postcondition in case of failure	The user gets informed about the reasons for failure (e.g. missing information).
Actor(s)	Economic operator Contracting authority User of the European VCD System External system (Nb: also called “user” in the following)
Initiating event	The user requests the download of the ontology (selecting menu item or link in European VCD System).
Description of interaction procedure with European VCD System	The user requests the download of the ontology. The system prepares the ontology for download, e.g. merging of different parts of the ontology (top-level, European and national ontologies). The system provides the ontology in downloadable form (in n3 notation or rdf/xml notation). The user downloads the ontology.
Extension(s)	--

Table 3-4: Use case description

The activity diagram in Figure 3-8 depicts the scenario described above.

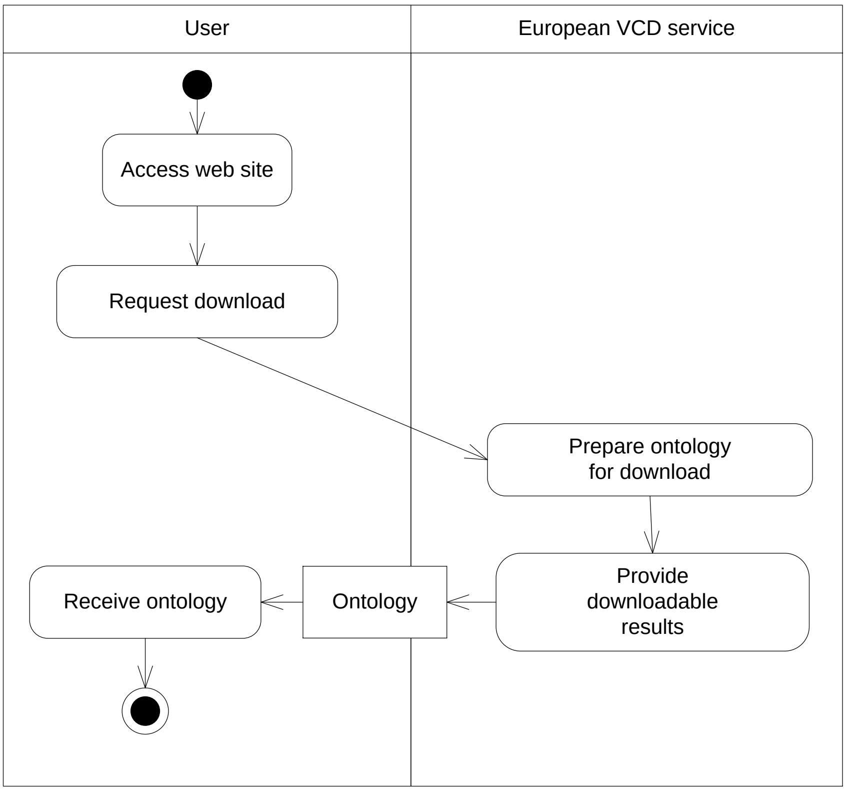


Figure 3-8: Activity diagram

3.6 Interaction Scenarios involving the European VCD System

Since the Interaction of the European VCD System with the user respectively the logical integration of functionality of the European VCD System into the National VCD System is vital for implementing the Interfaces, the following sections describe those two interaction scenarios in more technical detail. Successively, the Service Interface of the European VCD System (ESP Service Interface) is described in more detail-

- Sequence diagrams of interaction scenario 1: Fully automated Interaction between European VCD System and National VCD System

This section describes the interaction scenario drafted in chapter "**Feil! Fant ikke referansekilden.**". Furthermore some major components of the European VCD System are introduced and the data flow between them is also sketched in the sequence diagram.

The following components of the European VCD System are shown in **Feil! Fant ikke referansekilden.** and will therefore be described briefly:

ESP Service Interface:
This interface is responsible for interacting with the National VCD Systems, as described below.

User Interaction Service:
The User Interaction Service builds the graphical Interface to interact with the user. It acts as an "intelligent" Service within the osSso Machine and is assembled on the fly according to the underlying Ontology.

Transformer:

This component transforms the XML-input data (e.g. the VCD Pre-Skeleton Container) into its semantic (graph-) representation (RDFS) and vice versa. The VCD (Skeleton) Packager makes use of this component.

Reasoning Services:
The Reasoning Services enhance the input graph via applying the underlying rule set (represented in the ontology). Several Reasoners are operating within the European Service Provider (e.g. the Rule Based Reasoner and the OWL-DL Reasoner) at several stages. The reasoners are responsible for calculating the needed Evidences out of the input of the Economic Operator according to the Ontology.

osSso Machine:
The osSso Machine is an intelligent service assembly and execution layer, which detects missing information and fills in this missing data by calling the appropriate services in time. The User Interaction Service (and its fragments), the Reasoner Services, the Packager (wrapped Transformer) are examples of Services, orchestrated by the osSso Machine.

Ontology Manager:
The Ontology Manager handles the different data repositories and assembles the data to a virtual application graph, the whole European VCD System is operating upon. The Ontology Manager is also responsible for graph management functionality like persistency, querying and access control.

Application Controller:
The Application Controller is in charge of the program and data flow within the European VCD System.

Further Components:
A couple of further components act as background components for different tasks like Access and User Management, etc.

The components will be explained in greater detail in the next pages.

While the Economic Operator is interacting with the National VCD System, the selection of criteria and inference of possible evidences is performed by the European VCD System. This leads to a complex way of interaction between Economic Operator/User, National VCD System and European VCD System as depicted in Figure 3-9 below:

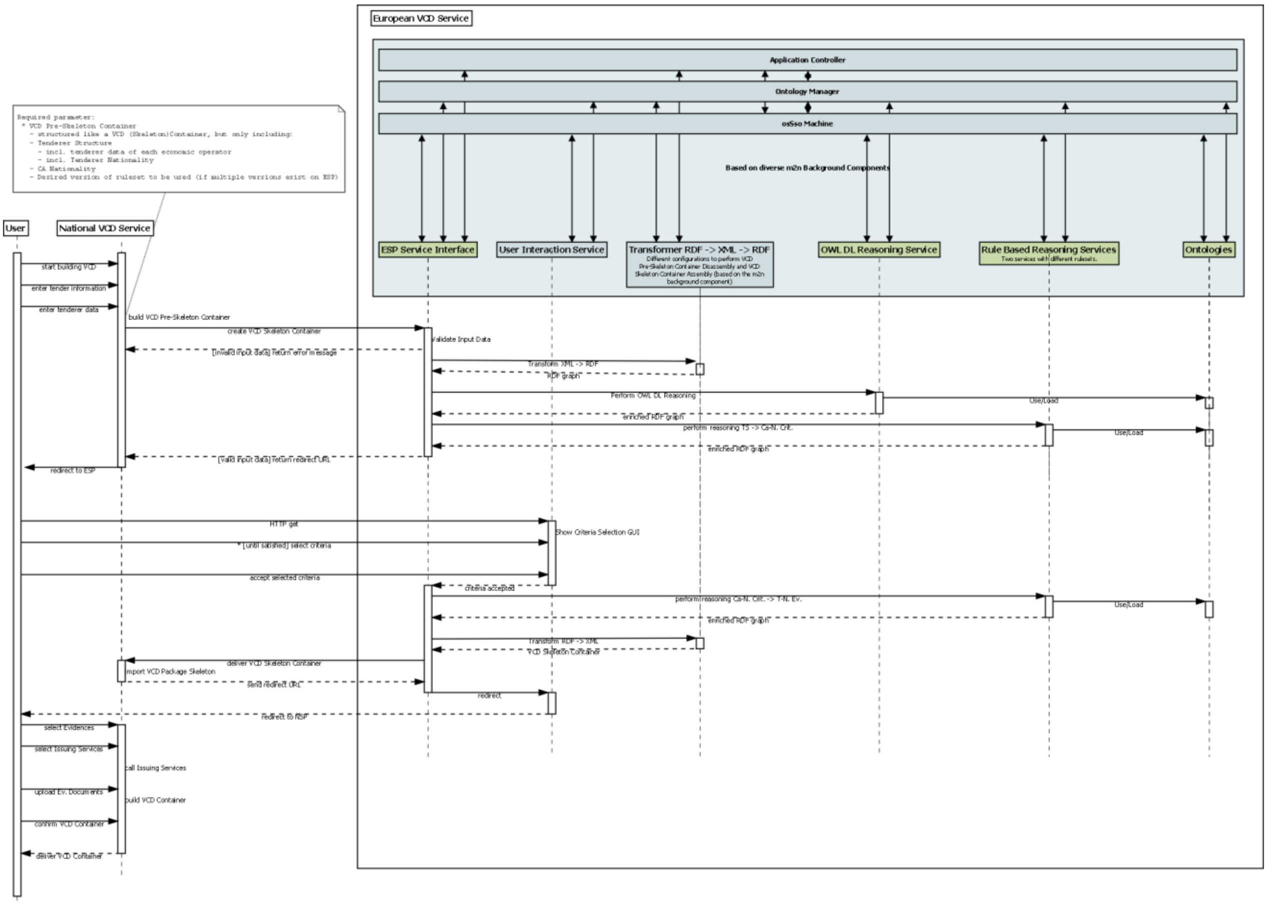

Required parameter:
* VCD Pre-Skeleton Container
- structured like a VCD (Skeleton)Container, but only including:
- Tenderer Structure
- incl. tenderer data of each economic operator
- incl. Tenderer Nationality
- CA Nationality
- Desired version of ruleset to be used (if multiple versions exist on ESP)
User
National VCD Service
European VCD Service
Application Controller
Ontology Manager
osSso Machine
Based on diverse m2n Background Components
ESP Service Interface
User Interaction Service
Transformer RDF -> XML -> RDF
Different configurations to perform VCD Pre-Skeleton Container Disassembly and VCD Skeleton Container Assembly (based on the m2n background component)
OWL DL Reasoning Service
Rule Based Reasoning Services
Two services with different rulesets.
Ontologies
start building VCD
enter tender information
enter tenderer data
build VCD Pre-Skeleton Container
create VCD Skeleton Container
Validate Input Data
[invalid input data] return error message
Transform XML -> RDF
RDF graph
Perform OWL DL Reasoning
Use/Load
enriched RDF graph
perform reasoning TS -> Ca-N. Crit.
Use/Load
enriched RDF graph
[valid input data] return redirect URL
redirect to ESP
HTTP get
Show Criteria Selection GUI
* [until satisfied] select criteria
accept selected criteria
criteria accepted
perform reasoning Ca-N. Crit. -> T-N. Ev.
Use/Load
enriched RDF graph
Transform RDF -> XML
VCD Skeleton Container
deliver VCD Skeleton Container
import VCD Package Skeleton
send redirect URL
redirect
redirect to NSP
select Evidences
select Issuing Services
call Issuing Services
upload Ev. Documents
build VCD Container
confirm VCD Container
deliver VCD Container

Figure 3-9: Sequence Diagram User-National VCD System-European VCD System

The basic flow of control contains the following steps:

- The Economic Operator enters all necessary data to generate the VCD Pre-Skeleton Container (e.g. Tenderer structure and Data and Contracting Authority nationality) at “his” national National VCD System (e.g. an Austrian Economic Operator turns to the Austrian National Service Provider).
- The National VCD System assembles the data into a VCD Pre-Skeleton Container
- The National VCD System sends this VCD Pre-Skeleton Container to the European VCD System, along with the address for the callback and a token to identify this request.
- The European VCD System analyses the received data and replies with an error message in case of error, or with a redirect-URL in case of success.
- In case of success, the National VCD System redirects the User to the URL returned by the European VCD System.
- The European VCD System calculates the suggested criteria and displays them to the user. The user interacts with the European VCD System to select/confirm the criteria to be proven for the individual Tenderer Structure Elements.
- When the Economic Operator is satisfied with the criterion selection, the European VCD System calculates the possible Evidence and creates a VCD Skeleton Container.
- The European VCD System sends the VCD Skeleton Container to the National VCD System, using the callback address and token supplied by the National VCD System in step 2. The National VCD System replies with a redirect URL.
- The European VCD System redirects the Economic Operator to the redirect URL from step 7.
- The Economic Operator now interacts with the National VCD System again.
- The National VCD System collects Evidence Documents and further Data by calling national Issuing Services and ...

- ... is finally passing the VCD Container back to the Economic Operator

Therefore, interactions between the National VCD System and European VCD System take place at steps 3 and 7. For the interaction in step 3, the European VCD System acts as a service and the National VCD System as its client. In step 7, the National VCD System acts as a service and the European VCD System as its client.

The service interface of the European VCD System in step 3 is named “European VCD System Interface”, the service the National VCD System is providing in step 7 is named “National VCD System Interface”.

- Sequence diagrams of interaction scenario 2 and 3: Direct Interaction between European VCD System and Economic Operator

This section explains the interaction scenario of the Economic Operator with the European VCD System.

In both of these scenarios, the user directly interacts with the European VCD System as depicted in Figure 3-10 below:

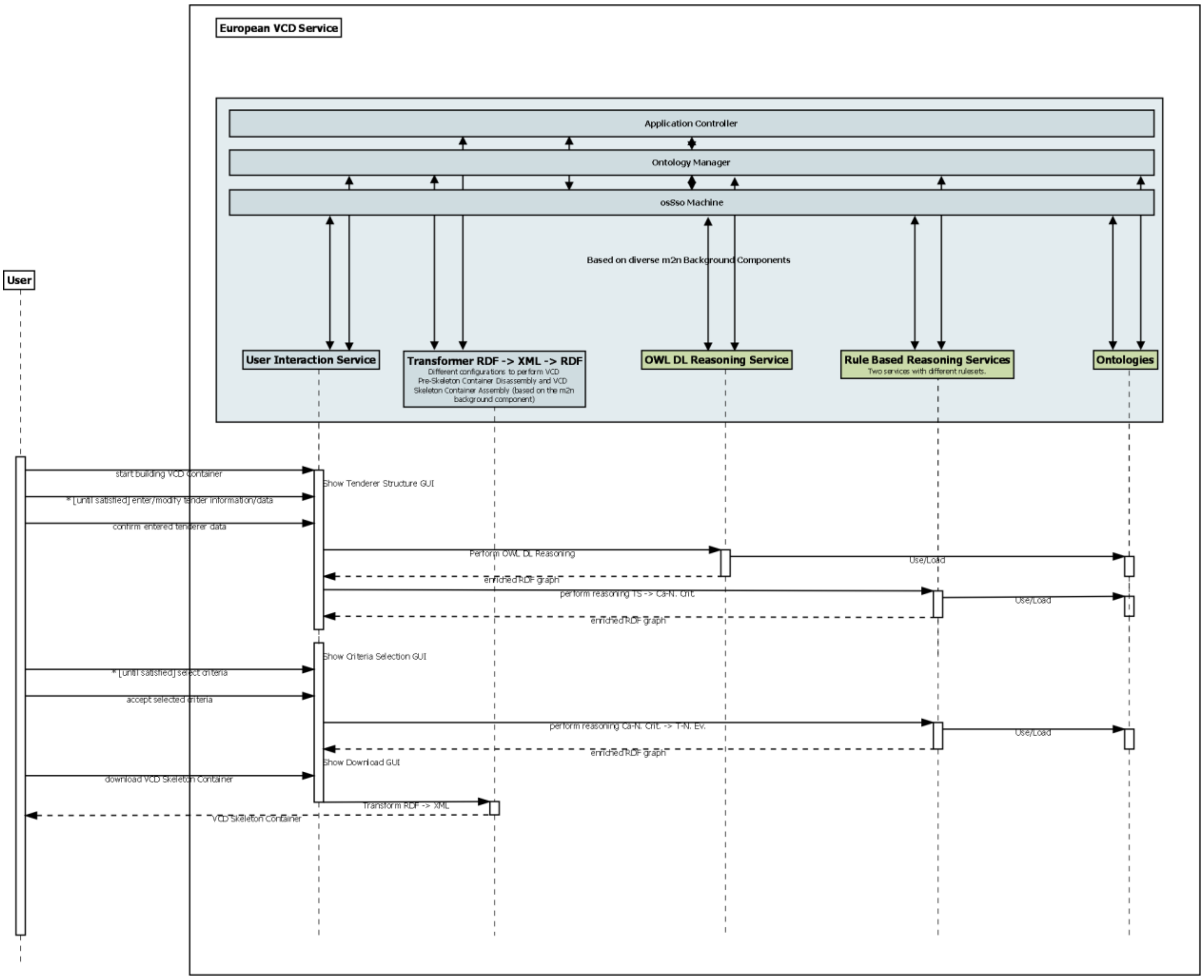

European VCD Service
Application Controller
Ontology Manager
osSso Machine
Based on diverse m2n Background Components
User
User Interaction Service
Transformer RDF -> XML -> RDF
Different configurations to perform VCD Pre-Skeleton Container Disassembly and VCD Skeleton Container Assembly (based on the m2n background component)
OWL DL Reasoning Service
Rule Based Reasoning Services
Two services with different rulesets.
Ontologies
start building VCD container
Show Tenderer Structure GUI
* [until satisfied] enter/modify tender information/data
confirm entered tenderer data
Perform OWL DL Reasoning
Use/Load
enriched RDF graph
perform reasoning TS -> Ca-N. Crit.
Use/Load
enriched RDF graph
Show Criteria Selection GUI
* [until satisfied] select criteria
accept selected criteria
perform reasoning Ca-N. Crit. -> T-N. Ev.
Use/Load
enriched RDF graph
Show Download GUI
download VCD Skeleton Container
Transform RDF -> XML
VCD Skeleton Container

Figure 3-10: Direct User Interaction with the European VCD System (without National VCD System)

- The Economic Operator enters all necessary data required by the European VCD System (e.g. Tenderer structure and Data and Contracting Authority's nationality) directly at the European VCD System.
- The European VCD System calculates the suggested criteria and displays them to the user.
- The Economic Operator interacts with the European VCD System to select/confirm the criteria to be proven for the individual Tenderer Structure Elements.
- When the Economic Operator is satisfied with the criterion selection, the European VCD System calculates the possible evidence, creates a VCD Skeleton Container and allows the Economic Operator to download it.
- The Economic Operator downloads the VCD Skeleton Container and manually or with the help of other applications obtains evidence documents to finalize his VCD Container.

The ESP Service Interface

This section is referring to a ZIP file containing XML-based formal machine-interpretable specifications of the interfaces involved and the expected input and output data. This ZIP file can be accessed via http://www.peppol.eu/work_in_progress/wp2-virtual-company-dossier/vcd-artefacts-1.

The system interfaces of the National VCD System and the European VCD System are applications of

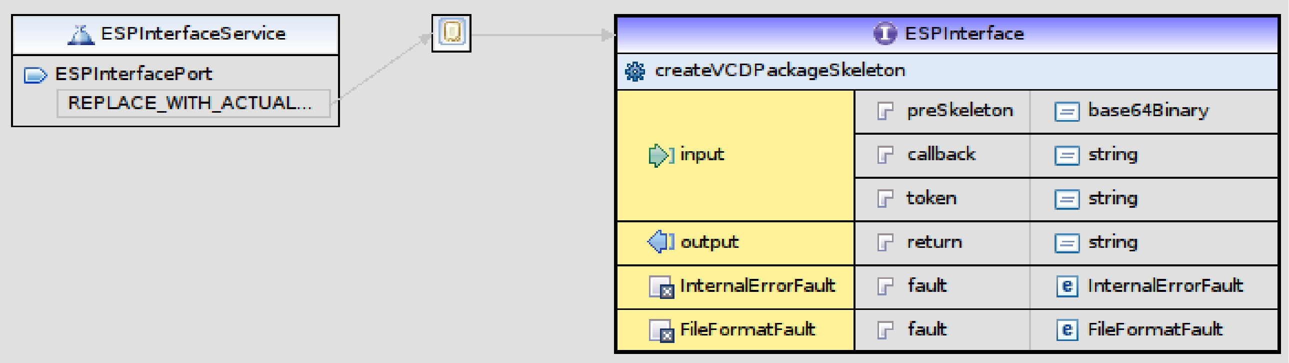
- SOAP 1.1 and
- WSDL 1.1

SOAP Interfaces

The SOAP Interfaces for the National VCD System Interface ("NSP Interface") and European VCD System Interface ("ESP Interface") are included in the directory WSDL in the ZIP file http://www.peppol.eu/work_in_progress/wp2-virtual-company-dossier/vcd-artefacts-1.

ESP
The ESPInterfaceService WSDL defines the Port Type ESPInterface which consists of one operation, createVCDPackageSkeleton. The operation takes as input a message made up of

- the base64-encoded VCD Package Pre-Skeleton,
- the URL of the NSP Interface used for the second interaction and
- a token used to authenticate the ESP against the NSP in that later interaction. This token should be unique, not easily guessable and not too long (< 1kB).



input	preSkeleton	base64Binary
	callback	string
	token	string
output	return	string
InternalErrorFault	fault	InternalErrorFault
FileFormatFault	fault	FileFormatFault

Figure 3-11: ESPInterface

The output of the operation can either be

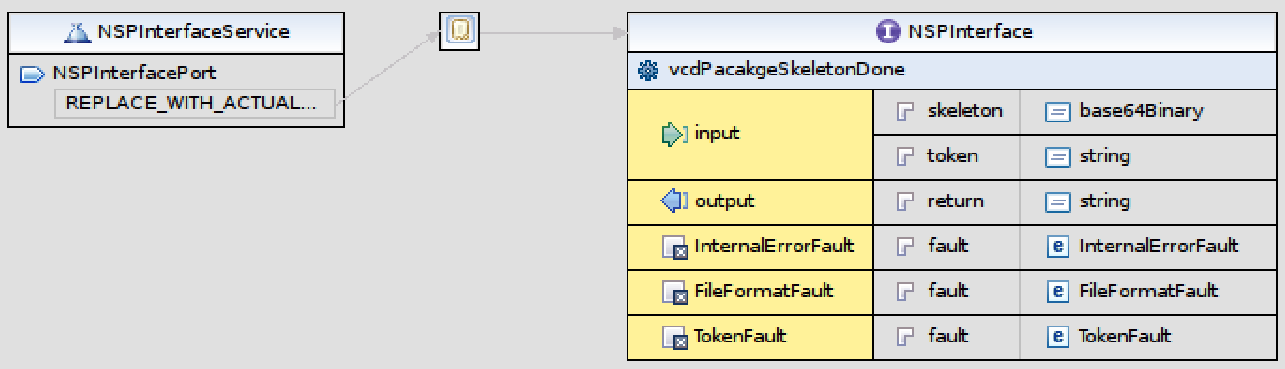
- a string containing the URL to be used for the redirection described in step 4 a FileFormatFault signalling a problem interpreting the VCD Package Pre-Skeleton, or
- an InternalErrorFault signalling an internal error.

The NSP calls createVCDPackageSkeleton to initiate the User-ESP interaction described in steps 3-8.

NSP

The NSPInterfaceService WSDL defines the Port Type NSPInterface, which consists of one Operation, vcdPackageSkeletonDone. The operation takes as inputs:

- the VCD Skeleton Container generated by the ESP and
- the token passed in the call to createVCDPackageSkeleton that initiated the processing that generated that VCD Skeleton Container.



NSPInterfaceService

NSPInterfacePort

REPLACE_WITH_ACTUAL...

NSPInterface

vcdPacakgeSkeletonDone

input	skeleton	base64Binary
	token	string
output	return	string
InternalErrorFault	fault	InternalErrorFault
FileFormatFault	fault	FileFormatFault
TokenFault	fault	TokenFault

Figure 3-12: NSPInterface

The output of the operation can either be

- a string containing the URL to be used for the redirection described in step 8,
- a FileFormatFault signalling a problem interpreting the VCD Skeleton Container,
- a TokenFault signalling that the National VCD System cannot accept the supplied token, or
- an InternalErrorFault signalling an internal error.

The European VCD System calls vcdPackageSkeletonDone after the user is done selecting criteria and is satisfied with the results. This is the end of the direct interaction between the user and the European VCD System. The call corresponds to step 7

<u>TokenFault</u>: If the National VCD System cannot accept the supplied token because it doesn't correspond to any token it has issued and not gotten a callback for yet, it answers the call with a TokenFault. The European VCD System should notify its administrator because this is probably caused by a programming error in either the National VCD System or the European VCD System or by some third party successfully impersonating the European VCD System. There is nothing that the National VCD System or European VCD System can do to rectify this situation without human investigation concerning the incident.

<u>Common Faults</u>: The two SOAP Faults defined are to be used in both interactions, so we define them in terms of service and client.

<u>FileFormatFault</u>: The service replies with FileFormatFault if the format of the supplied VCD Package (Pre-)Skeleton isn't as expected. The client should probably notify its administrator as this implies a programming error in either the client's export component or the service's import component. There is probably nothing that either the client or service can do to rectify this situation without human investigation concerning the incident.

<u>InternalErrorFault</u>: The service replies with an InternalErrorFault if there is an internal error that prevents correct processing at the moment. Depending on the severity of the problem, the most useful strategy for the client would be to retry the interaction after a reasonable waiting time, giving up after a to be defined number of retries with increasing waiting times between them. The InternalErrorFault can optionally contain an element retryAt, specifying the earliest point in time when the interaction should be retried.

Examples of SOAP messages

The following examples include HTTP header data some of which will depend on the actual HTTP client and server. The Base64 encoded binary parts are shortened with ellipsis.

createVCDPackageSkeleton request

```
POST http://localhost:8080/ESPInterface HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: ""
User-Agent: Jakarta Commons-HttpClient/3.1
Host: localhost:8080
Content-Length: 186200

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://service.peppol.m2n.at">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:createVCDPackageSkeleton>
      <callback>http://localhost:8080/NSPInterface</callback>
      <token>esp_CA03FE12BA55BE</token>
      <preSkeleton>UEsD…BBQA</preSkeleton>
    </ser:createVCDPackageSkeleton>
  </soapenv:Body>
</soapenv:Envelope>
```

createVCDPackageSkeleton successful reply

```
HTTP/1.1 200 OK
Content-Type: text/xml;charset=UTF-8
Transfer-Encoding: chunked
Server: Jetty(8.0.0.M0)

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:createVCDPackageSkeletonResponse xmlns:ns2="http://service.peppol.m2n.at">
      <return>http://m2n.at/ESP/redirect/135123</return>
    </ns2:createVCDPackageSkeletonResponse>
  </S:Body>
</S:Envelope>
```

createVCDPackageSkeleton FileFormatFault reply

```
HTTP/1.1 500 Server Error
Content-Type: text/xml;charset=UTF-8
Transfer-Encoding: chunked
Server: Jetty(8.0.0.M0)

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <S:Fault xmlns:ns4="http://www.w3.org/2003/05/soap-envelope">
      <faultcode>S:Server</faultcode>
      <faultstring>Not a ZIP file</faultstring>
      <detail>
        <ns2:PreSkeletonFormatFault xmlns:ns2="http://service.peppol.m2n.at">
          <message>Not a ZIP file</message>
        </ns2:PreSkeletonFormatFault>
      </detail>
    </S:Fault>
  </S:Body>
</S:Envelope>
```

vcdPackageSkeletonDone request

```
POST http://localhost:8080/NSPInterface HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: ""
User-Agent: Jakarta Commons-HttpClient/3.1
Host: localhost:8080
Content-Length: 345

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://service.peppol.m2n.at">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:vcdPacakgeSkeletonDone>
      <skeleton>UEsD…BBQA</skeleton>
      <token>esp_CA03FE12BA55BE</token>
    </ser:vcdPacakgeSkeletonDone>
  </soapenv:Body>
</soapenv:Envelope>
```

vcdPackageSkeletonDone successful reply

```
HTTP/1.1 200 OK
Content-Type: text/xml;charset=UTF-8
Transfer-Encoding: chunked
Server: Jetty(8.0.0.M0)

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:vcdPacakgeSkeletonDoneResponse xmlns:ns2="http://service.peppol.m2n.at">
      <return>http://m2n.at/NSP/redirect/031255</return>
    </ns2:vcdPacakgeSkeletonDoneResponse>
  </S:Body>
</S:Envelope>
```

4 Format of the VCD Package Pre-Skeleton and VCD Package Skeleton Files

Due to reasons of readability, the main concepts of the data-container passed between the National and the European VCD Systems are repeated in this section.

The VCD Skeleton Container and VCD Pre-Skeleton Container are based on VCD Container as specified by the PEPPOL WP2 VCD Schema Taskforce (see Figure 4-1 for overview):

The VCD Container (and by extension the VCD Skeleton Container and VCD Pre-Skeleton Container) show a simple physical structure: a ZIP file with a single XML file in the root folder (the VCD Package XML File) and a single subfolder for each contained VCD, which in turn contains the VCD XML File and additional files containing binary data, e.g. evidence documents.
The physical structure is the same for VCD Container, VCD Skeleton Container and VCD Pre-Skeleton Container, but the schemas for the XML files differ between the three.

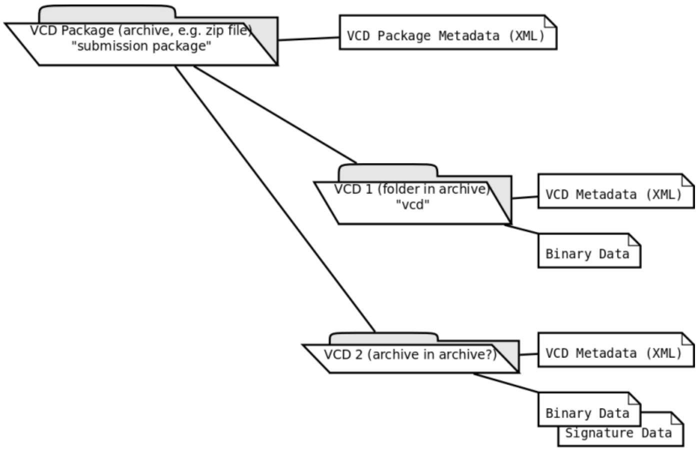

VCD Package (archive, e.g. zip file)
"submission package"
VCD Package Metadata (XML)
VCD 1 (folder in archive)
"vcd"
VCD Metadata (XML)
Binary Data
VCD 2 (archive in archive?)
VCD Metadata (XML)
Binary Data
Signature Data

Figure 4-1: Overview of VCD container

The VCD Skeleton Container Schema is the same as the VCD Container Schema, but Evidence Document Groups are not allowed to include Contained Documents. The elements "Proven by" and "proves" show semantics different to their semantics in the VCD Container Schema: they indicate the *possibility* to prove (evidence A, B or C can be used to prove Criterion D).

A patch to produce VCD Skeleton Container XSD Schema files from the included VCD Container XSD Schema files is included as file *VCD Package Skeleton.patch* in the directory VCD_Package_Schema in the accompanying ZIP file .

The VCD Pre-Skeleton Container Schema is the same as the VCD Container Schema, but no Criteria and no Evidence Document Groups are allowed.

A patch to produce VCD Pre-Skeleton Container XSD Schema files from the included VCD Container XSD Schema files is included as file *VCD Package Pre-Skeleton.patch* in the directory VCD_Package_Schema in the accompanying ZIP file.

While the European VCD System has to accept any valid VCD Pre-Skeleton Container for processing, the European VCD System is free to discard any data that it doesn't need for its operation.

For example, if the VCD Pre-Skeleton Container contains a DescriptiveNonEvidence for the EconomicOperator of one of its VirtualCompanyDossiers, the European VCD System is not required to include this data in the generated VCD Skeleton Container.

4.1 Use case group “EuropeanVCD Subsystem”

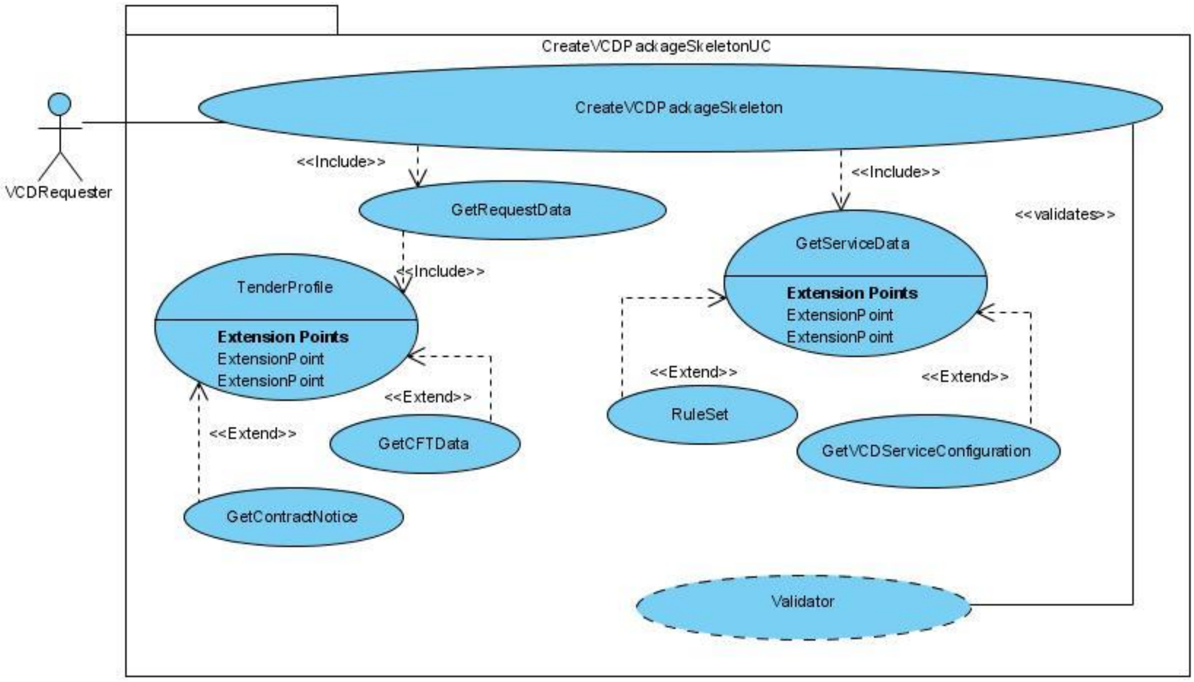


Figure 4-2: CreateVCD Skeleton Container Use-Case

Skeleton Packager Use-Case descriptions:

The use case drives the creation of an instance of the “VCDPackageskeleton”. The resulting document is capable to represent the qualification criteria defined by the contracting authority and the list of evidences resulting from the execution of the Pre_VCD mapping process. Besides, a document containing the mapping tree (reference to be added) can also be included.
The generated component, i.e. the “VCDPackageskeleton”, serves different purposes. At first, it gives a preliminary show of what the mapping activity has produced. Secondly, it offers a blue–print for the production of the true VCD and also the VCDPackage.

Readers should be aware that the “VCDPackageskeleton” can reach different content levels. This makes the validation mechanism different from Stage to Stage and from Step to Step.
The “VCDPackageskeleton” will contain a limited set of evidence data in its early status, including data that turn to be mandatory in a final VCD.
The physical shape of the “VCDPackageskeleton” is an XML instance that can be validated against the VCD package schema.

Aspect	Description
Objective	To create a preliminary and partial instance of a VCD package, i.e. the VCD skeleton container. It results from an interaction between a VCD Requester and the European VCD System System.
Results (postconditions in case of success)	A VCDPackageSkeleton(XML) instance
Precondition	Request data presented by the requester; Service data present within the system; Requester Identity validated; Rule set correctly identified and selected; Configuration steps correctly executed and completed by the requester
Postcondition in case of failure	Error message returned
Actor(s)	Economic Operator, Contracting Authority, European VCD System (self contained components), VCD systems
Initiating event	Request for a “VCD skeleton container”.
Description of interaction procedure with VCD Service (standard run)	Extends the mapping function defined by the European VCD sub-system.
Description of interaction procedure with VCD Service (alternative runs)	Delivery can directly reach the VCD Requesters or moved to VCD systems through a system supported interface.
Extension(s)	Allows National VCD System to Euro VCD system interactions
LEG/ORG	LEG: the mapping function requires legal support ORG: workflow model must adapt to the different stages, skeleton validation follows ORG directions.
Used in stage	Stage 1 as starting adoption, extensions and adaptations are required to use upper stages

Table 4-1:

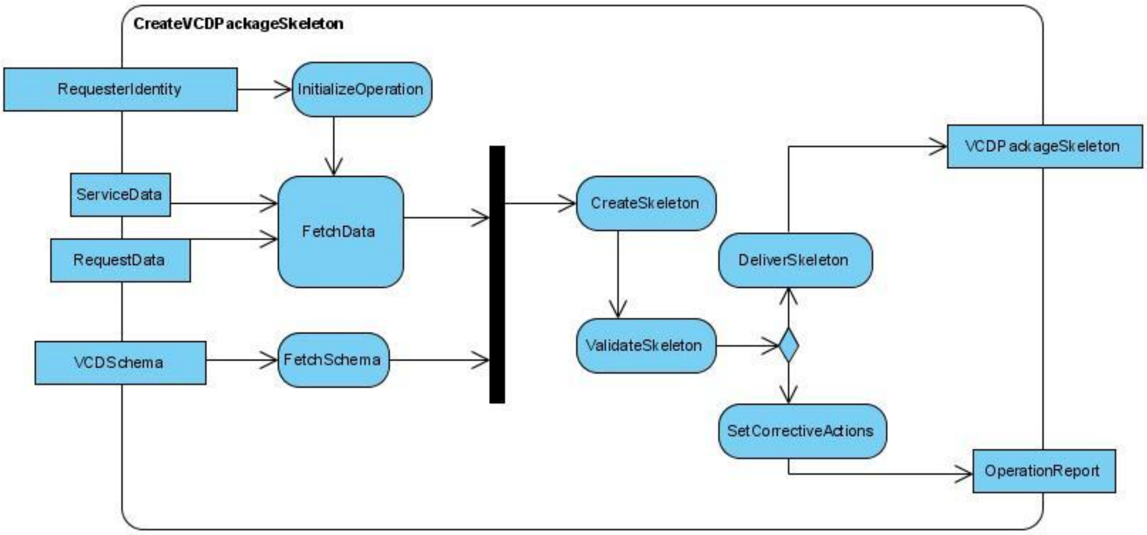


Figure 4-3: Create VCD skeleton container Activity & Actions

- VCD skeleton container creation

The VCD skeleton container contains both the qualification criteria defined by the contracting authority and the list of evidences resulting from the execution of the mapping process. Besides, a document containing the mapping tree can also be included.

The overall functionality of this component is the same as the one used in the VCD services. However, the validation mechanism is different, as the skeleton does not contain evidence data, hence, a number of mandatory fields remain empty.
The physical shape of the VCD skeleton container is an XML instance that can be validated against the VCD package schema.
The creation of the output of this component is divided into three steps:

- Creation of the VCD skeletons (one for each Economic Operator)
- Creation of the VCD skeleton container (for one tender context)
- Assembling of these artefacts into a VCD container

No identifier but 3 VCD schemas with different completeness levels: one for full VCD package, one for VCD skeleton container, one for VCD package pre-skeleton (tenderer structure only)

4.2 Functions supporting user interaction

- Dynamic intelligent forms

In order to react very quickly to changes in the underlying rule set and to give ultimate user support, the user interface should include dynamic intelligent forms for input-submission by the user, for instance:

- the selection of the country of origin of the contracting authority (“target country”)
- the input of information related to the tenderer structure (including legal form of organisation(s)/economic operators)
- the selection of the country of origin for each economic operator
- the selection / confirmation of qualification criteria and the visualisation of suitable evidences according to the ontology
- other relevant user input

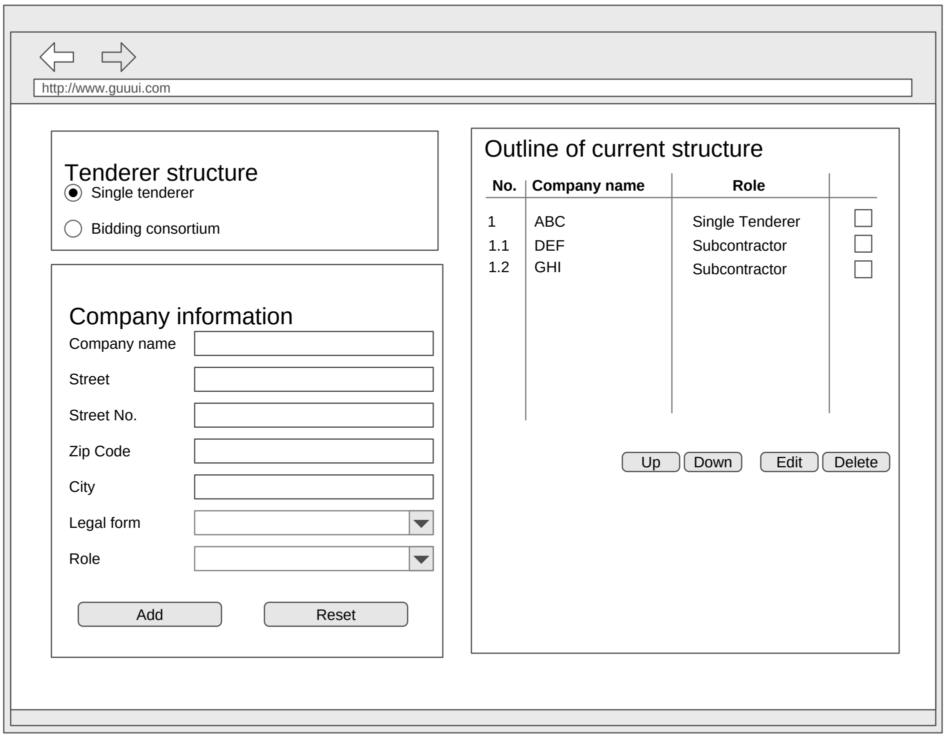
The form should only display relevant information respectively relevant input-options to the user - e.g. the list of qualification criteria - depending on the user’s choice of the target country. The forms need to be generated dynamically based on the currently existing ontology/rule set and based on the specific interaction step / the missing information. The interface for making the user select the criteria and/or evidences must be flexible in order to, on the one hand, avoid information overload and, on the other hand, collect exactly those information that the VCD Ontology Interaction Tool needs in order to generate the evidence suggestion and the mapping tree

- Input of tenderer information

The system must provide an interface for the input of tenderer information such as the tenderer structure (single tenderer/bidding consortium), the name, legal form and country of each company.
As shown in the mock up, the interface may consist of three areas. The first one (upper left part of the screen) provides means for choosing the tenderer structure, being either “Single Tenderer” or “Bidding Consortium”. Companies are added to the structure using the form in the second area (lower left part of the screen). Different information are to be provided by the user for each company such as name, legal form, country and the company’s role in the tenderer structure.

The relationship between contractor and subcontractor can be created using the field “Subcontractor of”. It is to be noted that this field is only displayed in case a firm acts as a subcontractor, i.e. the value in the field role is set to “subcontractor”.

The right side of the screen provides a list of companies added to the tenderer structure. The companies contained in that list can be edited (button “Edit”), deleted (button “Delete”) and their position in the list can be changed (buttons “Up”/”Down”). The numbering of the companies indicates possible contractor/subcontractor relationships, e.g. company 1.1 is a subcontractor of the first company. This is only an example of the input form which shall be altered in the course of implementation.



Tenderer structure

● Single tenderer

○ Bidding consortium

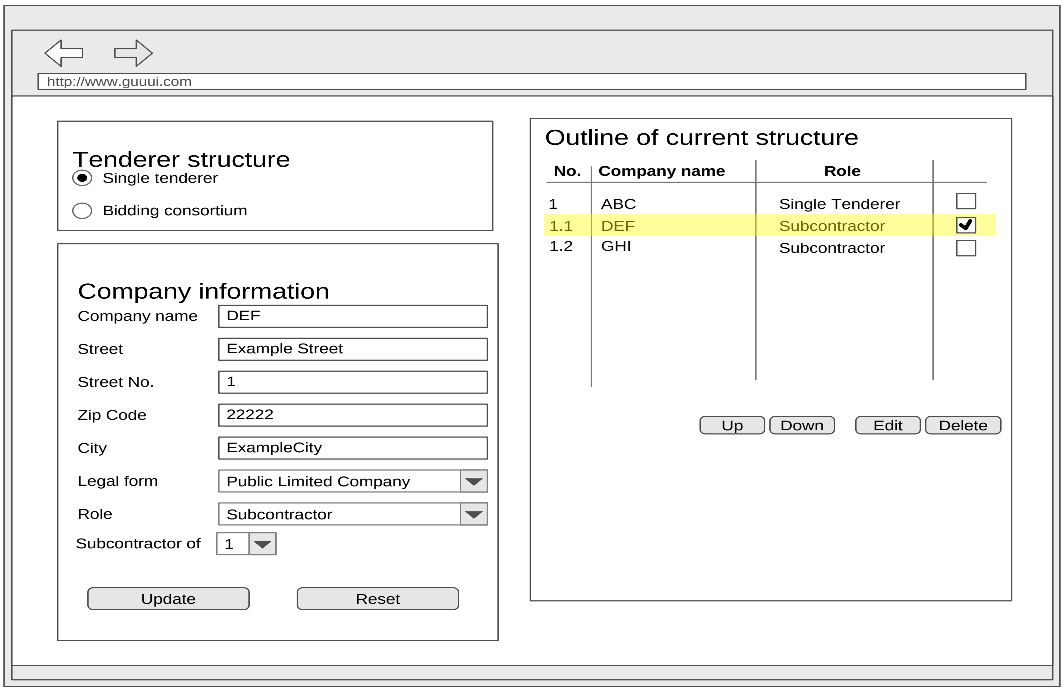
Company information

Company name	
Street	
Street No.	
Zip Code	
City	
Legal form	
Role	

Outline of current structure

No.	Company name	Role	
1	ABC	Single Tenderer	□
1.1	DEF	Subcontractor	□
1.2	GHI	Subcontractor	□

Figure 4-4: Mock-up describing the manual input of tenderer information



Tenderer structure

● Single tenderer

○ Bidding consortium

Company information

Company name	DEF
Street	Example Street
Street No.	1
Zip Code	22222
City	ExampleCity
Legal form	Public Limited Company
Role	Subcontractor
Subcontractor of	1

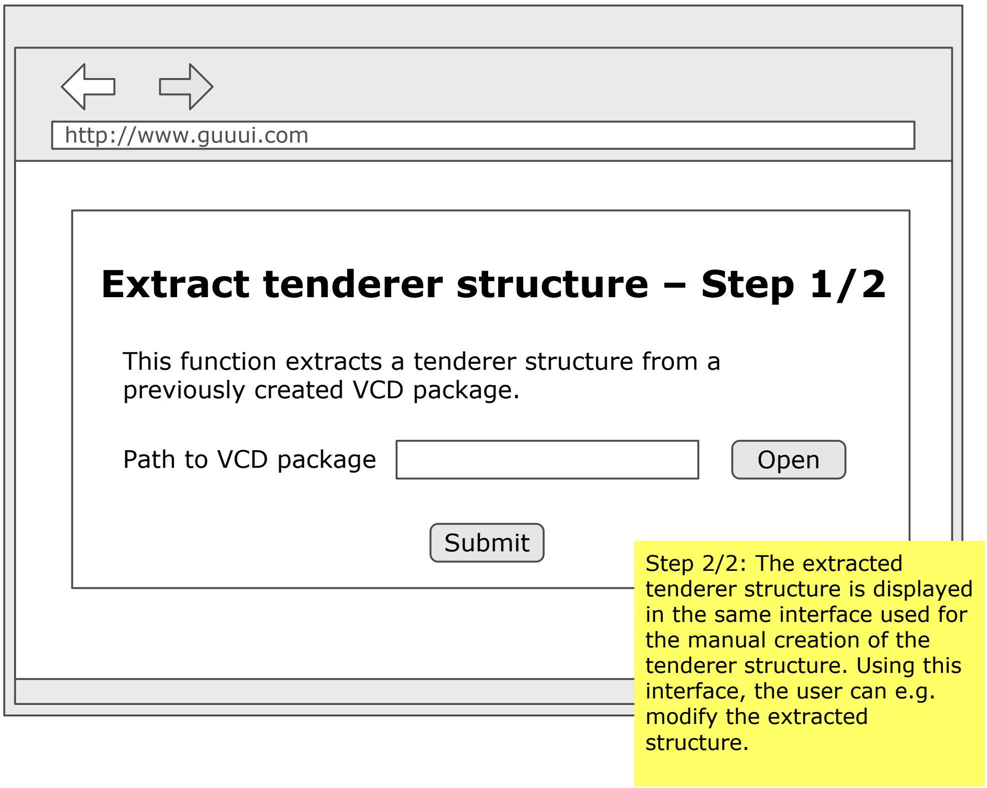
Outline of current structure

No.	Company name	Role	
1	ABC	Single Tenderer	□
1.1	DEF	Subcontractor	☑
1.2	GHI	Subcontractor	□

Figure 4-5: Mock-up showing the user interface for the editing of the tenderer information

Extraction of tenderer information from VCD Skeleton Container:
According to the functionality, the system should be able to automatically extract relevant tenderer information from a previously created VCD package, omitting the need for a manual input of these data. Using the interface depicted in Figure 4-5, a user can specify the path to a locally stored VCD

Skeleton Container and upload the latter to the European VCD System. The extracted tenderer information is displayed in the same interface used for the manual input of these data, an example being shown in Figure 4-6. The user is also informed about the successful extraction of the data from the VCD Skeleton Container.



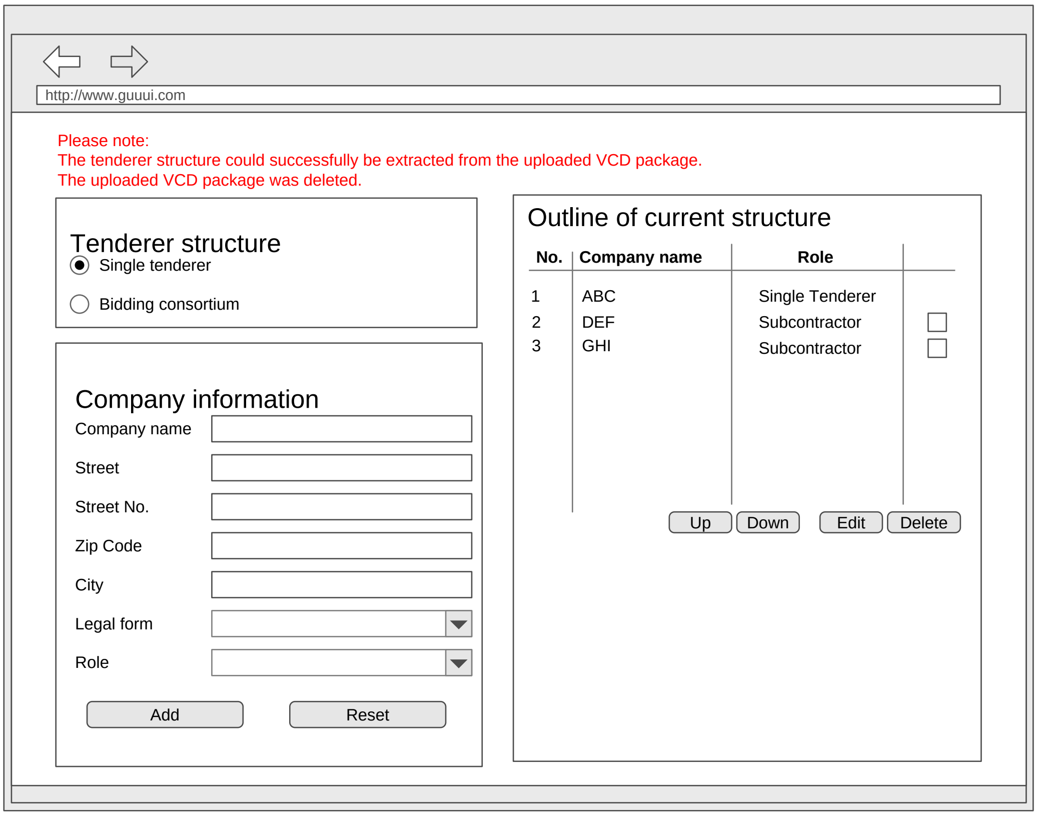
Extract tenderer structure – Step 1/2

This function extracts a tenderer structure from a previously created VCD package.

Path to VCD package

Step 2/2: The extracted tenderer structure is displayed in the same interface used for the manual creation of the tenderer structure. Using this interface, the user can e.g. modify the extracted structure.

Figure 4-6: Mock-up showing the user interface for the uploading of a previously created VCD (Skeleton) Container



Please note:
The tenderer structure could successfully be extracted from the uploaded VCD package.
The uploaded VCD package was deleted.

Tenderer structure

● Single tenderer

○ Bidding consortium

Company information

Company name

Street

Street No.

Zip Code

City

Legal form

Role

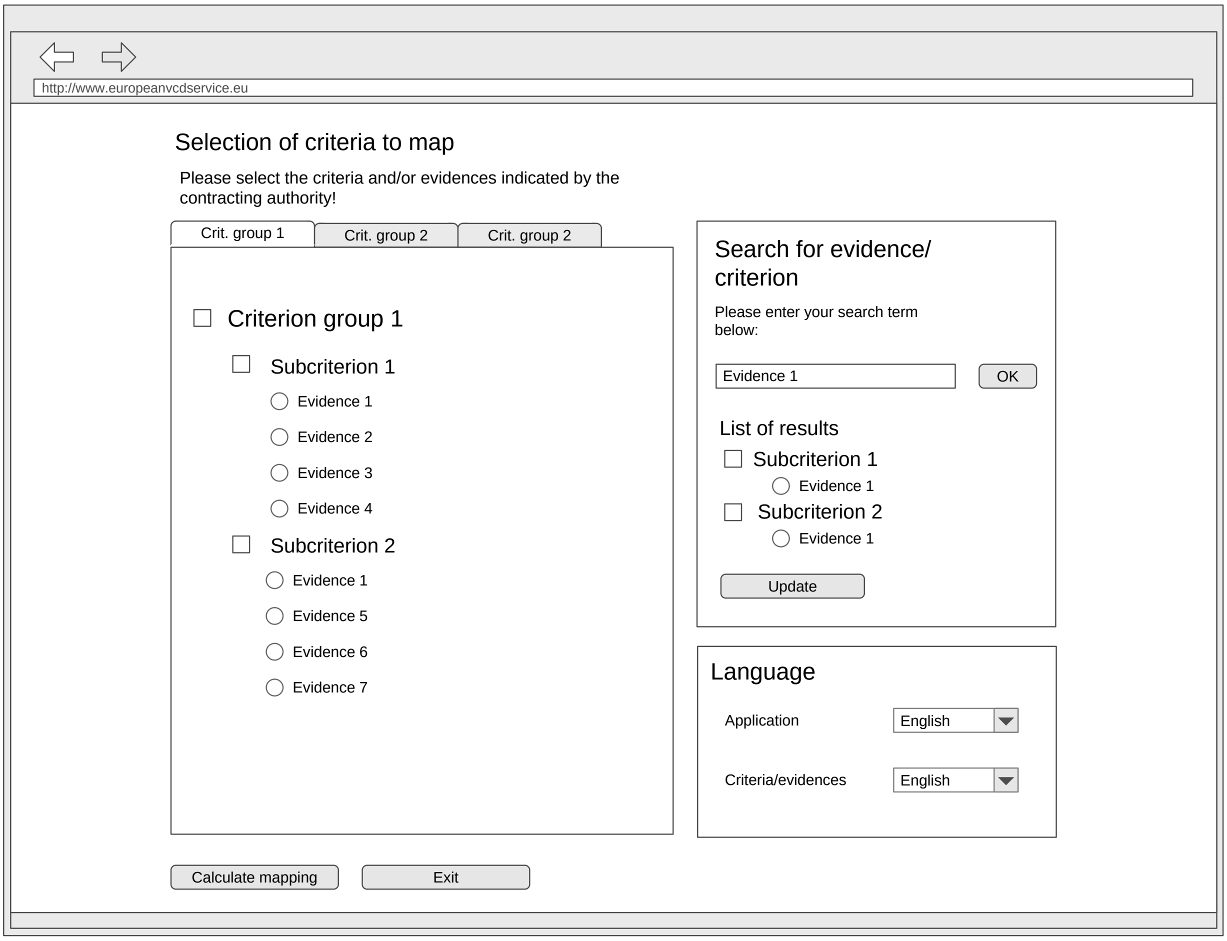
Outline of current structure

No.	Company name	Role	
1	ABC	Single Tenderer	
2	DEF	Subcontractor	□
3	GHI	Subcontractor	□

Figure 4-7: Mock-up showing the user interface for the modification of the data extracted from a previously created VCD Skeleton Container

- Criteria selection and Evidence information

Since the selection / confirmation of criteria demands a very good overview converning the different possibilities on the one hand and the suggestion of the system on the other hand, a tree view is proposed, structuring the suggested Contracting-Authority-National criteria top down grouping them according to the European criteria groups (articles). Since in some cases the contract notices name the evidences expected and not the criteria needed for mapping, the Contracting-Authority-National evidence names should be visualized as hints to the user. Apart from the tree-view, the criteria selection GUI should also provide the possibility to search for specific Contracting-Authority-national criteria and evidences. An example of a resulting interface to select the required criteria is shown in the figure below; however, different to the mockup, the system should already provide suggestions for fitting criteria according to the ontology.



Selection of criteria to map

Please select the criteria and/or evidences indicated by the contracting authority!

□ Criterion group 1
- □ Subcriterion 1
  - ○ Evidence 1
  - ○ Evidence 2
  - ○ Evidence 3
  - ○ Evidence 4
- □ Subcriterion 2
  - ○ Evidence 1
  - ○ Evidence 5
  - ○ Evidence 6
  - ○ Evidence 7

Search for evidence/ criterion

Please enter your search term below:

Evidence 1

List of results
- □ Subcriterion 1
  - ○ Evidence 1
- □ Subcriterion 2
  - ○ Evidence 1

Language

Application: English

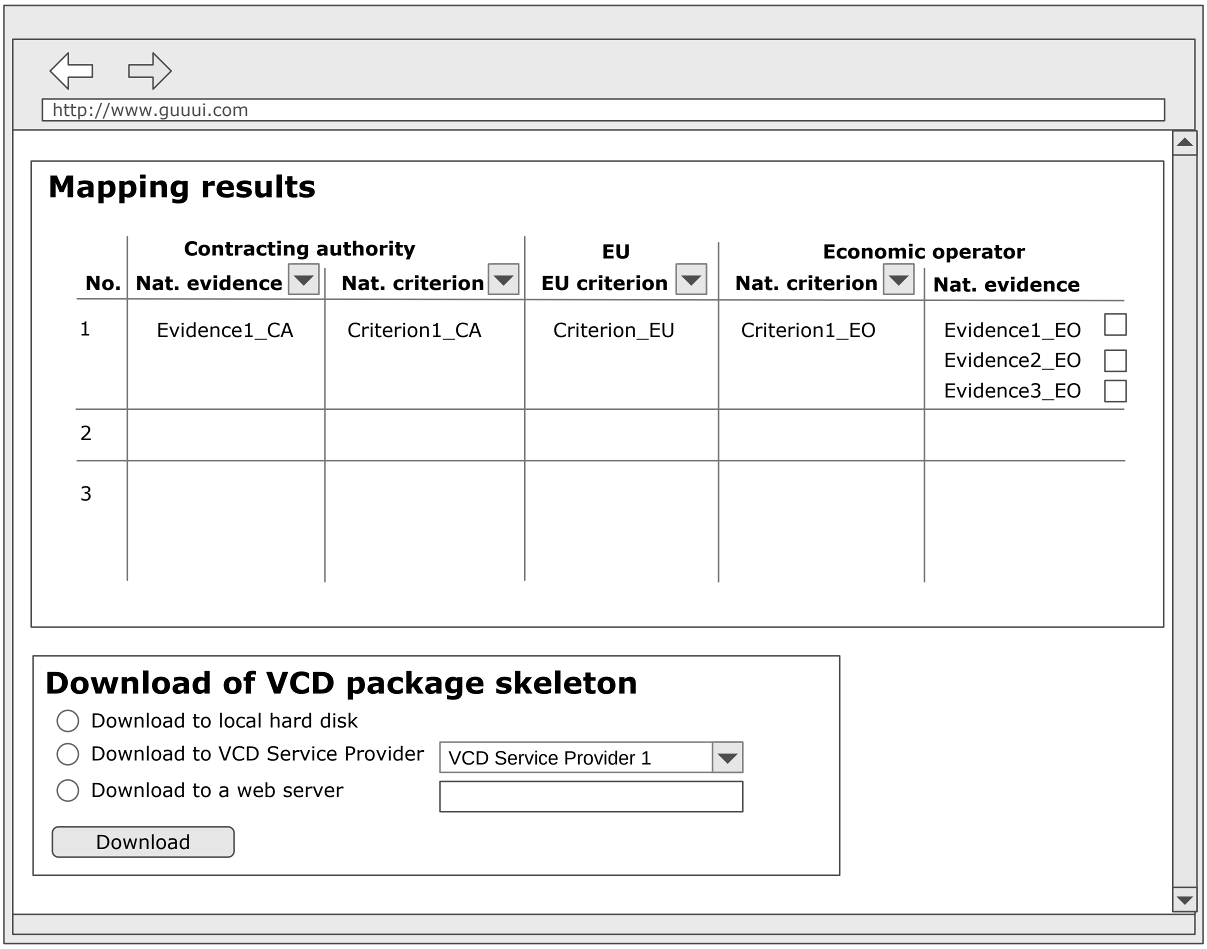
Criteria/evidences: English

Figure 4-8: Mock-up showing the user interface for the selection of criteria

After the user has entered the criteria, the system infers the evidence suggestion and displays it to the user.
For usability reasons, the interface informing the user regarding the list of evidences suitable to prove the suggested criteria, should possibly only include those criteria, that have been selected already. Since the user needs more information concerning the services providing the evidences in order to make a choice concerning which evidence to deliver, the evidence selection itself is part of the National VCD System or the VCD Editor. The VCD Skeleton Container will include all possible evidences to prove the selected criteria as described above.

In order to comprehensively view this “mapping information” (Contractring-Authority-National Criteria – EU Criteria – Tenderer-National-Criteria to Tenderer-National-Evidences, European VCD System should be able to visualise in an easy to understand and comprehensible way this mapping information generated from the system. Hence, it should at least contain four sections, describing the mappings:

- Evidences to national criteria (contracting authority) – optional (in case the call for tenders or contract notice lists evidences of the contracting authority)
- national criteria (according to nationality of contracting authority) to European criteria,
- European criteria to national criteria (according to nationality of economic operator) and
- national criteria (according to nationality of economic operator) to evidences.



Mapping results

	Contracting authority		EU	Economic operator	
No.	Nat. evidence	Nat. criterion	EU criterion	Nat. criterion	Nat. evidence
1	Evidence1_CA	Criterion1_CA	Criterion_EU	Criterion1_EO	Evidence1_EO □ Evidence2_EO □ Evidence3_EO □
2					
3					

Download of VCD package skeleton

○ Download to local hard disk

○ Download to VCD Service Provider VCD Service Provider 1

○ Download to a web server

Figure 4-9: Mock-up showing the mapping results

- Request generation of downloadable VCD skeleton container

After the user has seen the mapping information, he may request the generation of a downloadable VCD Skeleton Container. Hence, in the visualisation interface, a link or button will be shown to trigger this function.
The system then calls the VCD (Skeleton) Packager which generates the VCD Skeleton Container based on the inputted and derived dat.

- Download VCD Skeleton Container

After the generation of the VCD Skeleton Container, a download dialogue pops up and the user can choose the local directory the zip file will be downloaded

- Language selections for representations of interfaces and results

The user is able to choose from a number of languages available in the European VCD System. Two language selection options should be provided by the system: one for the selection of the language of the general user interface and one for the selection of the language of the criteria/evidences (English and language of the country of the contracting authority).

- Web form for ontology queries

The system should also provide an interface for the input of queries which are then executed by the European VCD Ontology Interaction Tool. To assist the user in the creation of such a query, a web form could provide a list of propositions, containing possible elements of interest.

.

5 Non-functional Requirements

5.1 Software Systems Attributes

The following non-functional specifications are common for all components of the European VCD System:

Extensibility: The European VCD System must be implemented as an extensible system, which can be upgraded and enlarged with new functionalities (e.g. implemented in new components) as intended to be added to the service. Likewise, it must allow enhanced by different external services interfaces such as e.g. the TED system or other national and European services to those mentioned in the specification document of the mapping component interact with the European VCD System without large efforts to implement such additional interfaces.

Reliability: Reliability is a result of the regulatory legal agreements for the European and national VCD services. It is formulated in service level agreements (SLAs) in order to produce services with agreed quality and reliability. The number of components of the different services determines the resources needed to deliver the defined reliability.
Different metrics exist to determine a system's "reliability". Sommerville proposes the following reliability metrics (Sommerville, 2007, p.209):
Probability of failure on demand (POFOD): The likelihood that the system will fail when a service request is made. For example, a POFOD of 0.001 means that 1 out of a thousand service requests may result in failure.
Rate of failure occurrence (ROCOF): The frequency of occurrence with which unexpected behaviour is likely to occur. A ROCOF of 2/100 means that 2 failures are likely to occur in each 100 operational time units. This metric is sometimes called the failure intensity.
Mean time to failure (MTTF): The average time between observed system failures. For example, an MTTF of 500 means that 1 failure can be expected every 500 time units.
Availability (AVAIL): The probability that the system is available for use at a given time. Availability of 0.998 means that the system is likely to be available for 998 of every 1,000 time units.
The classification of (Sommerville, 2007, p.211f) is followed to express the different failures that can occur in the different subsystems. For each failure, reliability metrics are specified for the piloting phase as well as for the production phase.
The failure classes indicate the type of failure according to Sommerville (Sommerville, 2007, p.211):

Transient	Occurs only with certain inputs
Permanent	Occurs with all inputs
Recoverable	System can recover without operator intervention
Unrecoverable	Operator intervention needed to recover from failure
Non-corrupting	Failure does not corrupt system state or data
Corrupting	Failure corrupts system state or data

Table 5-1: Failure classification

Accessibility: The system should be accessible to as many users as possible, including particularly those with disabilities. Design recommendations and guidelines for web content/interfaces are for example published by the Web Accessibility Initiative (WAI). The criteria defined by the WAI are recognized as de facto global standard for the design of accessible Web sites by the European Commission and thus have to be applied to all official web sites of the European Union. A respective communication entitled "eEurope 2002: Accessibility of Public Web Sites and their Content" details the usage of the WAI guidelines for the design of official web sites (see (Commission of the European Communities, 2001)).

Usability: The system has to be usable. According to Nielson (Nielsen, 1993, pp.26-37), usability encompasses five aspects: The system resp. its usage has to be easy to learn ("learnability"), it should be efficient to use ("efficiency"), its usage should be easy to remember ("memorability"), the user should commit as few errors as possible (“errors”) and the usage should be subjectively pleasing (“satisfaction”). Metrics have to be defined for each of these five aspects in order to specify concrete usability requirements (values), e.g.

Learnability: time a novice user needs to reach a specified level of proficiency in using the system (Nielsen, 1993, p.29).

Efficiency of use: time experienced users need to perform a number of defined tasks (Nielsen, 1993, pp.30-31).

Memorability: time a casual user (user who intermittently uses the system) needs to perform a set of defined tasks (Nielsen, 1993, pp.31-32).

Subjective satisfaction: Elaboration of a questionnaire to assess the user’s satisfaction. A possible metric would be the definition of an overall score resulting from the responses of the questionnaire. A certain value could then be specified, indicating the required/desired/needed level of satisfaction (Nielsen, 1993, pp. 33-37).

Performance – Response time: The system must be capable of delivering responses in a reasonable amount of time (low response time). An exact time frame has still to be defined. Nielsen (Nielsen, 1993, pp.135-137) states that a response time of 10 seconds is “about the limit for keeping the user’s attention focused on the dialogue”. In case of high response times, appropriate feedback has to be given to the user, e.g. in the form of a percentage indicator. A comparison of the European VCD System website with highly frequented websites could be performed in order to be able to determine a reasonable maximum response time.

Maintenance: The system should be maintainable. This includes for example “correcting errors which were not discovered in earlier stages of the life cycle, improving the implementation of system units and enhancing the system's services as new requirements are discovered” (Sommerville, 2007, p.67).

Legal compliance: The VCD System needs to meet all legal constraints in order to achieve acceptance by its users. This means that relevant regulatory specifications from participating Member States need to be listed and scrutinized so that legal compliance can be assured.
Legal compliance in a VCD Ontology Interaction activity, must be addressed by identifying some key entities, i.e. who is operating the service, which relationship is established between the entities affected by the service even in a non conventional business context. The actors model to be considered includes the European Service Provider, the requesters (EO, CA), the cooperating partners, i.e. the operators of the national VCD services. Addressing this legal compliance implies the definition of a service policy where the actions performed by the system are correctly announced, documented and inserted as terms of reference in a contract agreement. These documented actions are resulting into different generated objects or data sets. We can assume that the type of generated objects determines some corresponding responsibilities. These responsibilities may fall under country specific regulations that run beyond the European VCD System agreement. This last condition affects for instance the formation of database subject to data protection rules and the prevention of damages to these IT assets.
Security: -
In the definition and adoption of the European VCD system security principles we must stick to three basic rules:

- Find a balance between the use of our system assets and their protection: The role of the European VCD system is central to all the national implementations and this makes the system in a open network location. Accessibility should stay quite high, protection features not too cumbersome.

- Perform a Risk Analysis: Critical components have to be identified, in particular any component that bears long recovery actions, or components with compulsory protection obligations, i.e. data protection obligations or similar.
- Have a backup strategy and a recovery plan for the quickest re-activation of services: This must be considered the essential part of the system activation plan.

Actuality regarding changes in the ontology: The system should always be up-to-date and work with the most current version of the data (stored in the ontology).

Scalability: The system should be scalable, thus ensuring that "the capabilities of the system can be increased by increased by adding new resources to cope with new demands on the system" (Sommerville, 2007, p.267).

Testing / Conformance Assessment: All national VCD Systems will be connected to the European VCD System Provider (ESP), and the latter needs to offer testing and conformance assessment services in order to ensure full and correct interoperability within the network of national VCD SP and other services in the PEPPOL infrastructure. Such services need to be available for testing of new and up-dated applications, registers and code lists.
It needs to be decided whether conformance assessment of a version of an installation should be certified before it is accepted for inclusion in a production environment. The agreed quality level will constrain requirements to testing and conformance assessment, the tools, methods, procedures and their organization.
The testing will assess structure and content of exchanged files; the service will check that communicating application and its version, that the file structure is correct, that its code lists are known, and that it produces correct response to input of various test data.
The VCD systems will interface other services of PEPPOL, like the infrastructure and the signature systems; testing and conformance assessment covering all interfaces need to be available permanently. This implies that all testing and conformance assessment of PEPPOL systems need to be strongly coordinated.
The VCD systems will in due time also connect to the Publication Office's TED system by using an interface emerging from the current eSender (http://circa.europa.eu/Public/irc/opoce/eproc/library?l=/public/1_general&vm=detailed&sb=Title); the testing and conformance assessment for the interfacing is outlined in chapter 2 of http://circa.europa.eu/Public/irc/opoce/eproc/library?l=/public/1_general/qualification_schema206p/_EN_1.0_&a=d .
The testing and conformance assessment, including certification, of interfaces between VCD systems should be built and offered accordingly; it is crucial to offer testing of VCD SP systems when new versions of software and code lists are taken in use.

Multilingualism: The user interfaces of the components of the European VCD System need to be available in all official languages of the EU. For the runtime of the project, it is necessary that the user interfaces of the components of the European VCD System are available in the languages of all WP2 partners:

- French
- German
- Greek
- Italian
- Norwegian

5.2 GUI attributes

The European VCD System must provide support for multilingualism ("internationalization"). The following features have been identified:

Each concept, which is displayed in the user interface, must have an English representation (project requirement). Especially each national criterion and evidence will also have to be translated into English. This translation must be part of the Ontology. [4]
Each label which is displayed in the user interface must also be available in all languages used in the pilots, particularly in the following ones:

Table 5-3 Languages for the UI of the dynamic intelligent web form of the VCD Ontology interaction tool

Implementation partner	Language
All (Common project language)	English
Austria	German
France	French
Greece	Greek
Italy	Italian
Norway	Norwegian

Table 5-2: Languages fort he UI of the dynamic intelligent web form of the VCD Ontology Interaction tool

6 TED interface

6.1 Introduction

TED and national publication systems have interfaces that make them accept structured announcements from contracting authorities and deliver structured files to the systems used by economic operators. Contracting authorities find services on http://simap.europa.eu/ojs_esenders/list_of_ojs_esenders/index_en.htm, these are also rendering services to Economic Operators by alerting on CPV codes and giving guidance on how to reply.
More than 90 % of all notices sent to TED are in structured XML format according to DTD from more than sixty services in EEA. The CIRCA project of the Publication Office is working on the upgrading to XML Schema, which will be implemented and piloted by the end of 2010; public information about the project's results and progress is available from the TED eSender documentation folder on http://circa.europa.eu/Public/irc/opoce/eproc/library?l=/public/1_general&vm=detailed&sb=Title.
The documentation contains specification of XML Schemas for interfacing, further it specifies several identifiers that are of importance for VCD systems, and also methods and procedures for testing to become accepted partner of TED are detailed.
The eSender specifications cover presentation of fixed texts of the CFT form in all official EU languages (not Norwegian).
In order to interface TED the relevant VCD systems need match parts of the eSender XML Schema specifications. Further work therefore needs to be constrained by the referred documentation.

6.2 Overview

The general description has been used to derive Figure 6-1. The drawing shows how contracting authorities and economic operators are interoperating through TED; the parties are encircled in Figure 6-1.

[4] A national ontology and rule set must be available in English and the respective language(s) of the country -> requirement to be implemented in the ontology.

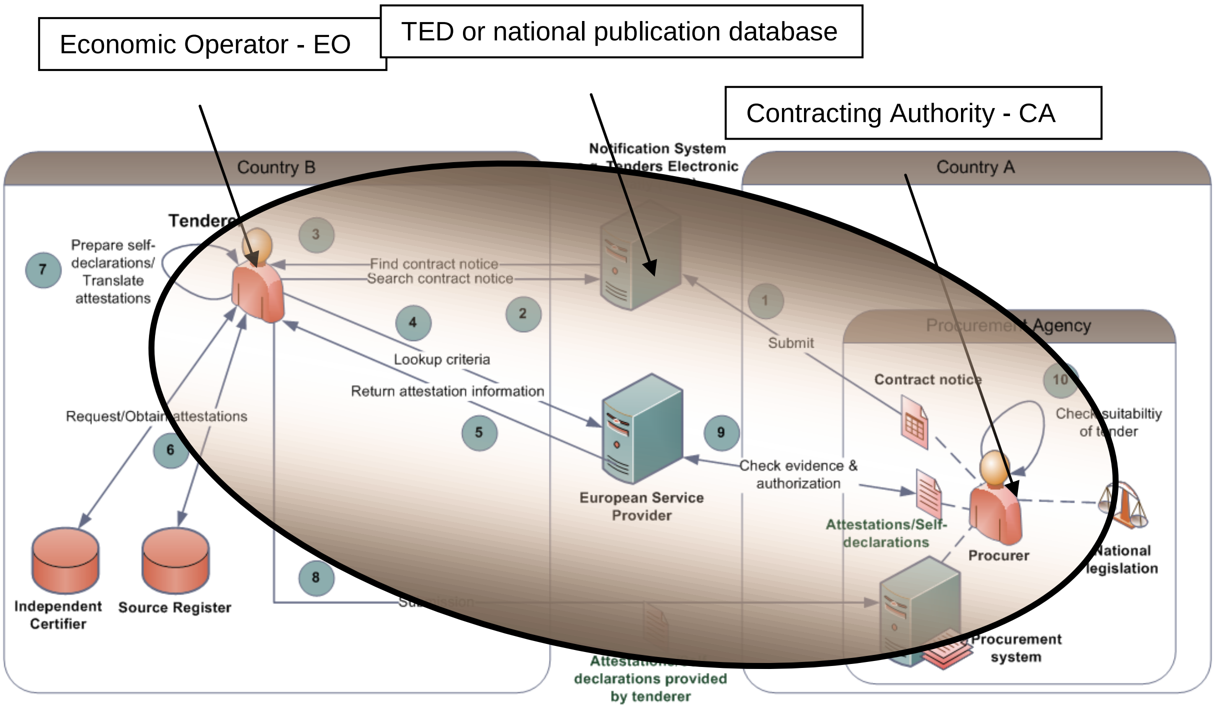

Economic Operator - EO
TED or national publication database
Contracting Authority - CA
Country B
Country A
Notification System
Tenderer
Prepare self-declarations/ Translate attestations
7
3
Find contract notice
Search contract notice
4
2
1
Submit
Procurement Agency
Lookup criteria
Return attestation information
Contract notice
10
Check suitabiltiy of tender
Request/Obtain attestations
6
5
9
Check evidence & authorization
European Service Provider
Attestations/Self-declarations
Procurer
National legislation
8
Independent Certifier
Source Register
Submission
Attestations/Self-declarations provided by tenderer
Procurement system

Figure 6-1: Both CA and EO need a structured TED interface

National publication databases will interface TED by using the eSender specifications for:

- Interfacing
- Sharing of data and identifiers
- Testing and Conformance Assessment
- Presentation in more official languages

Both main parties need to interface TED by use of its specifications; the VCD Service Provider will need to interoperate with Contracting Authority, Economic Operator and TED, hence the TED interface specification will become essential for interoperability within its environment and must be implemented by the VCD SPs to create its interoperability.

The figure clearly shows how systems of the contracting authority and the economic operator share information in TED, and in order to become an intermediary service the VCD SP systems also need to become a sharing partner by interfacing TED.

6.3 Activity Diagram

The activity diagram shows how the Economic Operator need a TED interface to retrieve data from the Call for Tender (CFT) in TED; some of the data retrieved will be used to specify the VCD skeleton, others to label it and create time line for the processing.

Having built the skeleton package the VCD SP will be able to collect the evidences form the Issuing Bodies (IB), and the construct the complete VCD package to be submitted to the Contracting Authority (CA).

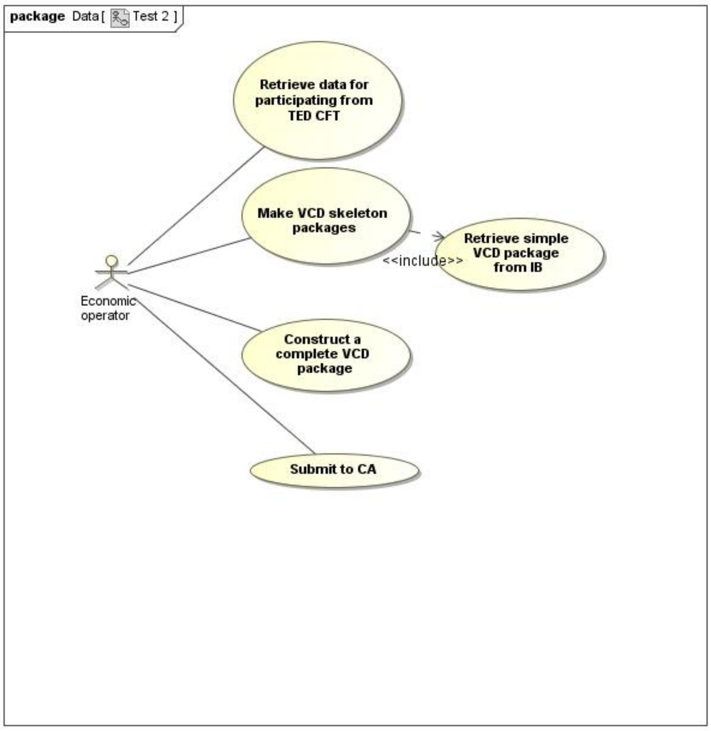

package Data [Test 2]
Retrieve data for participating from TED CFT
Make VCD skeleton packages
<<include>>
Retrieve simple VCD package from IB
Economic operator
Construct a complete VCD package
Submit to CA

Figure 6-2: Activity Diagram

6.4 Functions to be interfaced

- Overall description of the notification structure to be interfaced

Economic operators are using services that alert them when section II.1.5 of calls for tender contain given CPV codes. If the EO wants to prepare a response, the services extract information from header and sections in notices in TED:

- From header:
  - Date of publication
  - TED registration number
  - Type of contract
  - Type of procedure
  - Country and Region of contracting authority
  - Type of publication

- From section I.1:
  - Name of contracting authority – likely to deviate from the one in the business register
  - Address of contracting authority to be used for the tendering process – likely to deviate from the official one and the one used for other procurement processes.
  - Names, phone numbers and e-mail addresses of contact persons

- From section II.1.1:
  - Contracting authority’s reference number of tender – will also be used to label contract and as reference number later.
  - Contracting authority’s naming of the case and the contract.

- From section II.1.6:
  - CPV codes

- From section III.2.1:
  - Names of evidential documents, like Company registration certificate, Value added tax certificate, Self declarations.

- From section IV.1:
  - Type of procedure.

- From section IV.2.1:
  - Time-limit for receipt of tenders or request to participate

- From section IV.3.1:
  - Administrative information: File reference number attributed by the contracting authority

- From section IV.3.6:
  - Language in which tenders or requests to participate may be drawn up.
  - Minimum time frame during which the tenderer must maintain the tender: Duration on months.

- From section VI.2:
  - URL of additional information

The systems support the writing and inclusion of the tender documents by using the collected data, and they build up the entire file by attaching electronic copies of evidential documentation. The finished file will finally be made available for the contracting authority.

Full data models will belong to the eSender specifications, so far the following are available:

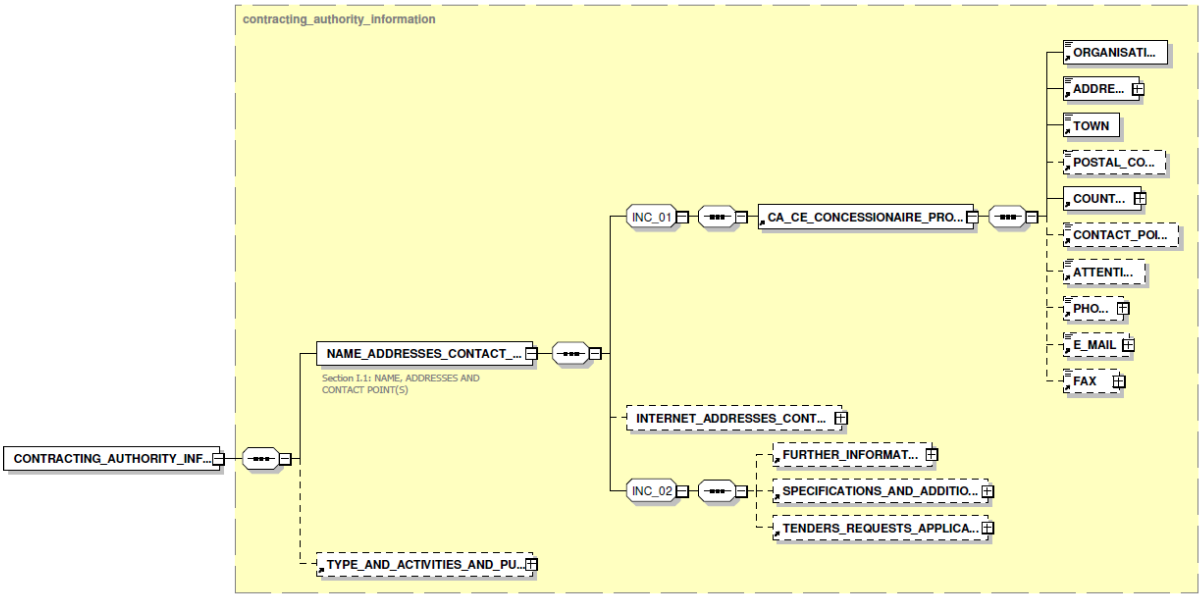

contracting_authority_information
CONTRACTING_AUTHORITY_INF...
NAME_ADDRESSES_CONTACT_...
Section I.1: NAME, ADDRESSES AND CONTACT POINT(S)
INC_01
CA_CE_CONCESSIONAIRE_PRO...
ORGANISATI...
ADDRE...
TOWN
POSTAL_CO...
COUNT...
CONTACT_POI...
ATTENTI...
PHO...
E_MAIL
FAX
INTERNET_ADDRESSES_CONT...
INC_02
FURTHER_INFORMAT...
SPECIFICATIONS_AND_ADDITIO...
TENDERS_REQUESTS_APPLICA...
TYPE_AND_ACTIVITIES_AND_PU...

Figure 6-3: Contracting Authority Information data model from eSender

There are many difficulties before to have a possibility to use the eSender XML message to Peppol purposes:
For the moment and it should be the same when the new schema will be in production (scheduled for end of 2010), the criterions (and requested evidences) are in a free text field and so are not structured to be computable.

The eSender XML message is an input message to TED not an output message (ie : not for the economic operator, there is an XML message (different from eSender XML message) for the “licence holders”, ie : companies who pay TED for this information in XML files format to build a subscription based service).

The new schema will be ready late for our purpose of using it for the pilots.

The only information we should have when the new schema will be in production with the proposed schema, is the tender information header and the CA information.

So that is why, we have proposed to use the fact than the CA are dealing with different tools to prepare their tender notices (CFT) to generate (through an evolution of their tools) a VCD pre-skeleton which will be joined to the list of tender documents. It could include all the information described in the VCD skeleton except the economic operator information.

In a far future, the same information could be retrieved automatically from an evolution of TED notices but the conditions are not met in 2010.

Index of Figures

Index of Tables

Abbreviations

PEPPOL Glossary of Terms and Abbreviations can be found in deliverable 7.3b