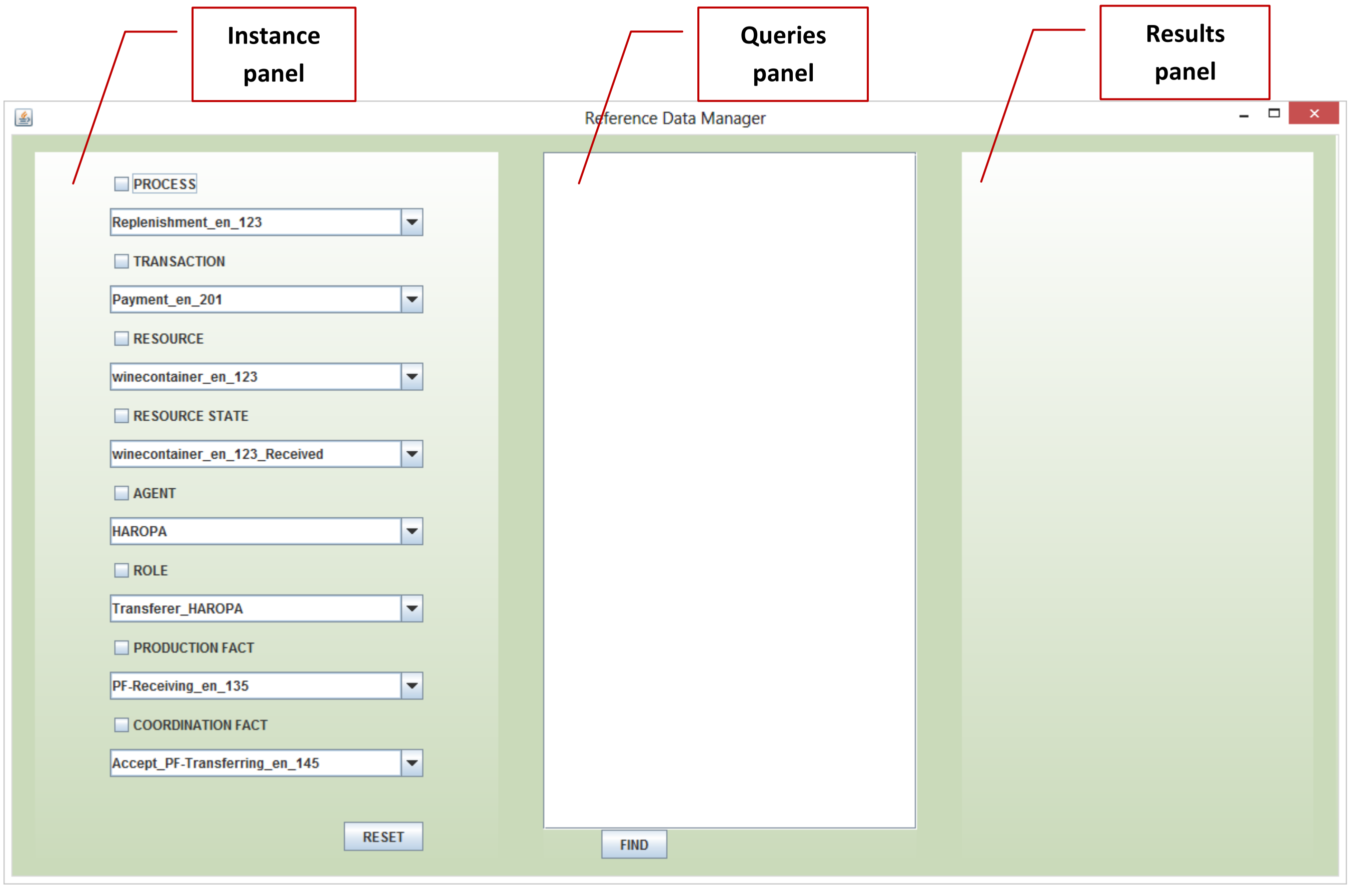
We present a software prototype **Reference Data Manager** for dealing with the knowledge base and extracting relevant information. The graphic user interface of the programming tool is shown in Figure 1 and Figure 2.

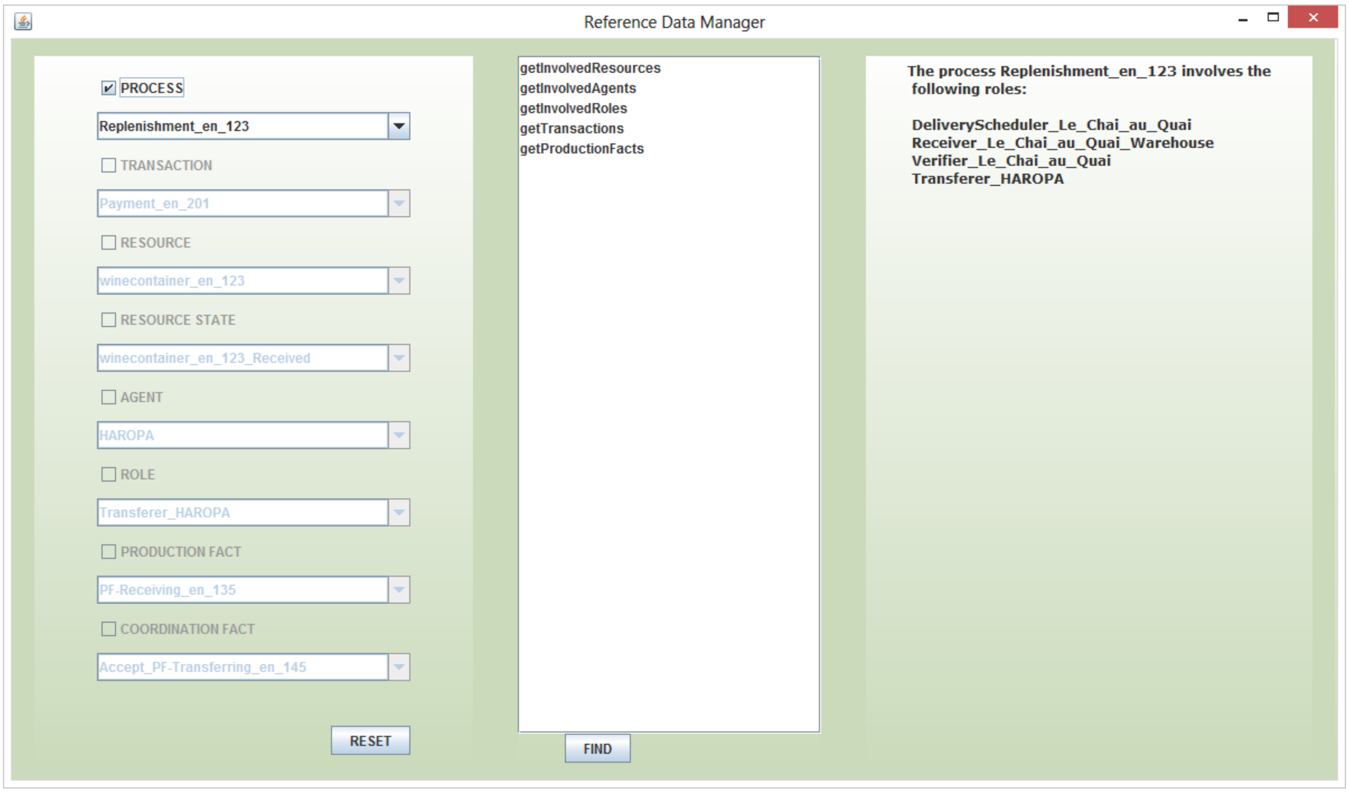
The main window contains three panels: the panel for choosing instances, the panel for choosing queries and the panel for displaying results (see Figure 1).



□ PROCESS
Replenishment_en_123
□ TRANSACTION
Payment_en_201
□ RESOURCE
winecontainer_en_123
□ RESOURCE STATE
winecontainer_en_123_Received
□ AGENT
HAROPA
□ ROLE
Transferer_HAROPA
□ PRODUCTION FACT
PF-Receiving_en_135
□ COORDINATION FACT
Accept_PF-Transferring_en_145

Figure 1.

The Instance panel contains different categories of instances that can be chosen for further information processing. Once the instance is chosen, available queries for that category appear on the Queries panel. The result will be shown on the Results panel after pressing the Find button (see Figure 2).



Reference Data Manager

☑ PROCESS

Replenishment_en_123

☐ TRANSACTION

Payment_en_201

☐ RESOURCE

winecontainer_en_123

☐ RESOURCE STATE

winecontainer_en_123_Received

☐ AGENT

HAROPA

☐ ROLE

Transferer_HAROPA

☐ PRODUCTION FACT

PF-Receiving_en_135

☐ COORDINATION FACT

Accept_PF-Transferring_en_145

getInvolvedResources
getInvolvedAgents
getInvolvedRoles
getTransactions
getProductionFacts

The process Replenishment_en_123 involves the following roles:

DeliveryScheduler_Le_Chai_au_Quai
Receiver_Le_Chai_au_Quai_Warehouse
Verifier_Le_Chai_au_Quai
Transferer_HAROPA

Figure 2.

The aim of Reference Data Manager is to allow users to deal with the knowledge base in the user-friendly way. Bellow there are some queries to the knowledge base using the special query language SPARQL. SPARQL is rather complicated for unprepared users. Users of the program are free from some technical complexities and they have possibilities to extract all relevant information.

1. What transactions does the process *Replenishment_en_123* include?

Query	Result

PREFIX fm: <*FrameworkNS*>
PREFIX pr: <*OperLevelNS*>
SELECT DISTINCT ?transaction
FROM <*OperLevelNS*>
WHERE {
?tuple fm:whole pr:Replenishment_en_123.
?tuple fm:temporalPart ?transaction.
}
	Payment_en_201
	SchedulingDeliveries_en_161
	Receiving_en_135
	Transferring_en_145
	Verification_en_143

2. Who is the initiator of the process *Transferring_en_145*? Or, in other words, who is responsible for the coordination fact *Request_PF-Transferring_en_145*?

Query	Result
PREFIX fm: <*FrameworkNS*> PREFIX pr: <*OperLevelNS*> SELECT DISTINCT ?responsibility FROM <*OperLevelNS*> WHERE { ?tuple fm:hasResponsible pr:Requset_PF-Transferring_en_145." ?tuple fm:hasResponsibility	Laithwaites_wine

?responsibility. }	

3. What transaction precedes the transaction *Verification_en_143?*

Query	Result

PREFIX fm: <*FrameworkNS*>
PREFIX pr: <*OperLevelNS*>
SELECT DISTINCT ?before
FROM <*OperLevelNS*>
WHERE {
?tuple fm:after pr:Verification_en_143.
?tuple fm:before ?before.
}
	Receiving_en_135

4. What are necessary conditions for the production fact *PF-Transferring_en_145*?

Query	Result

PREFIX fm: <*FrameworkNS*>
PREFIX pr: <*OperLevelNS*>
SELECT DISTINCT ?precondition
FROM <*OperLevelNS*>
WHERE {
?tuple fm:hasFact pr:PF-Transferring_en_145.
?tuple fm:hasPrecondition ?precondition.
}
	winecontainer_en_123_Verified

5. What roles are involved into the process *Replenishment_en_123*?

Query	Result

PREFIX fm: <*FrameworkNS*>
PREFIX pr: <*OperLevelNS*>
SELECT DISTINCT ?actorRole
FROM <*OperLevelNS*>
WHERE {
?tuple fm:whole pr:Replenishment_en_123.
?tuple fm:role ?actorRole.
}
	Receiver_Le_Chai_au_Quai_Warehouse
	Verifier_Le_Chai_au_Quai
	DeliveryScheduler_Le_Chai_au_Quai
	PaymentProcessor_Laithwaites_wine
	Transferer_HAROPA