

Req Type	Description	Purpose	Quality measure
essential	Repository interface	To provide access to the repository	The rule editor accesses the repository
essential	Graphical user interface	To enable the user to work in an environment he/she is used to	The user communicates with the rule editor via a graphical user interface
desired	Rule editor configuration	To configure the rule editor GUI layout and behavior to adapt to specific requirements	The GUI layout and the behavior of the rule editor can be configured .

5.1.2. Business ramblings editor

The editor allows a business analyst or some other domain expert to express business rules.

Req Type	Description	Purpose	Quality measure
important	Creation, modification and deletion of business ramblings	To manipulate business ramblings	The business analyst can create, edit and delete business ramblings
important	Interface to business vocabulary (glossary)	To specify business rules in accordance with terms in the business vocabulary	The business analyst can relate expressions in business ramblings to the business vocabulary
important	Rambling searching feature	To search business ramblings by search criteria	A user can enter search criteria. The user can navigate in the result set and select rules for editing.

5.1.3. Business rule editor (natural language-like rule representation)

The editor allows a user (business analyst or developer) to edit business rules.

Req Type	Description	Purpose	Quality measure
essential	Creation, modification and deletion of business rules	To manipulate business rules in a language that is easily understandable to a business analyst.	A user can create, modify and delete rules in the natural language-like representation
desired	Rule editing templates	To enable a business analyst to edit rules based on a template	A user can create a rule from a template and edit it
desired	Rule editing wizard	To enable a business analyst to edit rules guided by an syntax-sensitive editing wizard	The rule editor provides a syntax-sensitive wizard that helps the user to create business rules
essential	Editing of rule properties	To set properties for a rule	The user user can set properties for a rule
important	Rule searching feature	To search rules by search criteria	A user can enter search criteria. The user can navigate in the result set and select rules for editing.

essential	Syntax checking feature	To prevent a user from formulating syntactically erroneous rules.	The syntax checking facility is capable of detecting syntactic errors. Rule groups cannot be saved to the repository if errors have been detected.
essential	Consistency and integrity checking feature	To avoid inconsistencies or integrity violations in the repository.	The checking feature is capable of detecting consistency and integrity violations. Rule groups cannot be saved to the repository if errors have been detected.
important	Rule comparison	To enable a business analyst to compare two different versions of a rule in business-language representation	The differences between the two versions are shown in an easily recognizable form. Different colours are used to identify the differences (additions, deletions, modifications).
desired	Relationships to business ramblings	To establish relationships to business ramblings	The user can establish relationships to business ramblings
important	Impact analysis on new or evolving business rules	To make sure that new or evolving business rules can only be introduced after impact analysis	The rule editor displays business rules which have relationships to the rule being edited
desired	Dependency detection	To ensure that business rules with relationships to other rules can not be deleted without warning	The tool issues a warning message if the user attempts to delete a rule which has a relationship to another rule
important	Rule transformation	To transform a rule from natural language-like syntax to the native rule language syntax.	The tool performs automatic transformation from one representation into the other

5.1.4. Business rule editor (native rule language representation)

The editor enables a developer to perform rule editing in the tool-specific native rule language.

Req Type	Description	Purpose	Quality measure
essential	Creation, modification and deletion of business rules	To manipulate rules in the native rule language	The user can create, modify and delete rules in the native rule language
desired	Rule editing templates	To enable a business analyst to edit rules based on a template	A user can create a rule from a template and edit it
desired	Rule editing wizard	To enable a developer to edit rules guided by an syntax-sensitive editing wizard	The rule editor provides a syntax-sensitive wizard that helps the user to create business rules

essential	Editing of rule properties	To set properties for a rule	The user user can set properties for a rule
important	Rule searching feature	To search rules by search criteria	A user can enter search criteria. The user can navigate in the result set and select rules for editing.
essential	Syntax checking feature	To prevent a user from formulating syntactically erroneous rules	The syntax checking facility is capable of detecting syntactic errors
essential	Consistency and integrity checking feature	To avoid inconsistencies or integrity violations in the repository.	The checking feature is capable of detecting consistency and integrity violations. The rule group is not saved to the repository if errors have been detected.
important	Impact analysis on new or evolving business rules	To make sure that new or evolving business rules can only be introduced after impact analysis	The rule editor displays business rules which have relationships to the rule being edited
important	Rule comparison	To enable a developer to compare two different versions of a rule in native rule language representation.	The differences between the two versions are shown in an easily recognizable form. Different colours are used to identify the differences (additions, deletions, modifications).
desired	Dependency detection	To ensure that business rules with relationships to other rules can not be deleted without warning	The tool issues a warning message if the user attempts to delete a rule which has a relationship to another rule

5.2. Mapping editor

A mapping editor may be a separate tool or it may be integrated with the rule editor. The mapping editor lets users perform mappings between object types, operations and attributes specified in rules and in UML class diagrams.

5.2.1. UML mapping editor

A UML mapping editor accesses UML class diagrams and visualizes them in a window. Access is read-only. An interface to the CASE tool repository must be provided to accomplish integration.

Req Type	Description	Purpose	Quality measure
desired	Visualization of UML class diagram	To display UML class diagrams	The tool can display UML class diagrams
important	Mapping of class names	To map classes to entity types defined by the business analyst	The user can map classes to entity types

important	Mapping of method names	To map methods to operations defined by the business analyst	The user can map methods to operations
important	Mapping of parameters	To map method parameters to parameters defined by the business analyst	The user can map attributes to parameters

5.2.2. Class mapping editor

A class mapping editor accesses Java classes and visualizes them in a window. Classes are available after source code has been developed and compiled.

Req Type	Description	Purpose	Quality measure
important	Inspection of Java classes and interfaces	To enable the developer to directly access Java classes and interfaces	The developer can access Java classes and interfaces by using Java introspection
important	Mapping of class names	To map classes to entity types defined by the business analyst	The user can map classes to entity types
important	Mapping of method names	To map methods to operations defined by the business analyst	The user can map methods to operations
important	Mapping of parameters	To map method parameters to parameters defined by the business analyst	The user can map parameters to parameters

5.3. Debugger

A debugger is used at run-time. It enables the user to manually control the execution of an application by using specific debug commands.

5.3.1. Breakpoints

Breakpoints are used to have the rule engine stop execution at some specified object state or event. Processing can be resumed by command.

Req Type	Description	Purpose	Quality measure
important	Rule breakpoints	To enable a developer to set rule breakpoints	The developer can set rule breakpoints. Execution stops when a selected rule is about to be executed.
important	Class breakpoints	To enable a developer to set class breakpoints	The developer can set class breakpoints. Execution stops when any instance of a selected class is "touched" by the rule engine.
important	Object breakpoints	To enable a developer to set object breakpoints	The developer can set object breakpoints. Execution stops when the selected object is "touched" by the rule engine.

important	Object attribute breakpoints	To enable a developer to set object attribute breakpoints	The developer can set object attribute breakpoints. Execution stops when a selected attribute value is modified by a rule
important	Event breakpoints	To enable a developer to set event-specific breakpoints.	The developer can set event breakpoints. Execution stops when a selected event occurs.
important	Selective breakpoint removal	To selectively remove breakpoints	The developer can selectively remove breakpoints of various types (see above).

5.3.2. Watchpoints

Watchpoints are used to monitor object states or events at run-time. Processing does not stop as is the case with breakpoints. Specified object states and events are recorded during processing.

Req Type	Description	Purpose	Quality measure
desired	Rule watchpoints	To enable a developer to set rule watchpoints	The developer can set rule watchpoints. Status information is recorded when a selected rule is about to be executed.
desired	Class watchpoints	To enable a developer to set class watchpoints	The developer can set class watchpoints. Status information is recorded when any instance of a selected class is "touched" by the rule engine.
desired	Object watchpoints	To enable a developer to set object watchpoints	The developer can set object watchpoints. Status information is recorded when the selected object is "touched" by the rule engine.
desired	Object attribute watchpoints	To enable a developer to set object attribute watchpoints	The developer can set object attribute breakpoints. Status information is recorded when a selected attribute value is modified by a rule
desired	Event watchpoints	To enable a developer to set event-specific watchpoints.	The developer can set event breakpoints. Status information is recorded when a selected event occurs.
desired	Selective watchpoint removal	To selectively remove watchpoints	The developer can remove watchpoints.

5.3.3. Stepwise execution

Rule execution can be stopped after each rule . The user can inspect the dynamic memory (working memory) and/or the agenda.