Visure
the requirements company

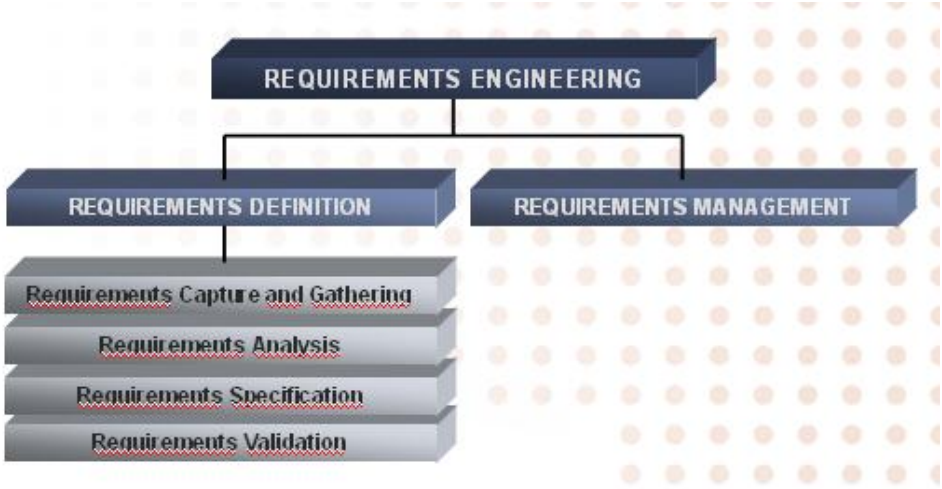
IRQA

"THE REQUIREMENTS SOLUTION"

IRQA "THE REQUIREMENTS SOLUTION"

DESCRIPTION OF SOLUTION

IRQA is an end-to-end Requirements Definition and Management solution that covers all the main Requirements Engineering activities. It has been specifically designed in order to provide state-of-the-art support to each of the practices related to the complete Requirements Management and Requirements Development process areas along the complete Development Lifecycle.



Additionally, IRQA supports out of the box, and without the need of other tools, some of the key additional activities related to Requirements Definition and Management, such as:

- Requirements Verification: Test Case Definition & Management.
- Requirements Modelling
- Requirements Definition office tools roundtrip import and export.
- Change Request Management.
- Requirements Metrics and Dashboard.
- Graphical workflow capabilities

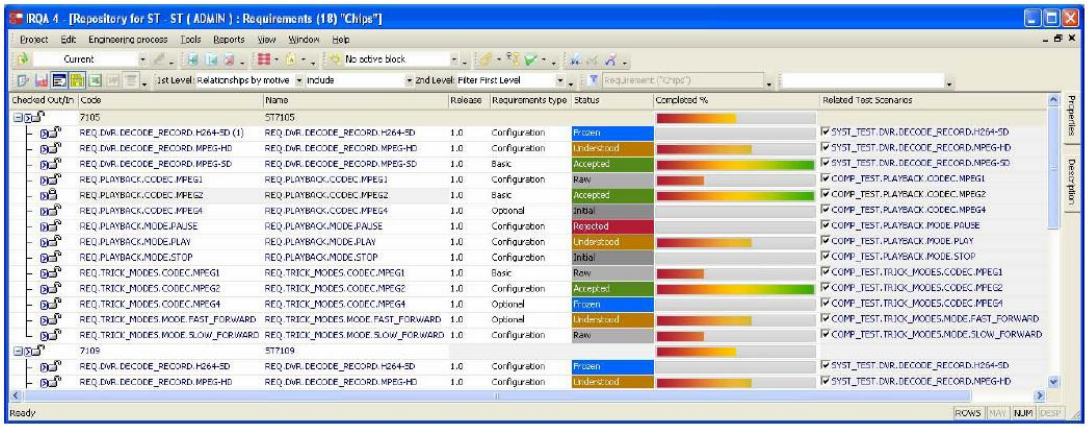
Some of the key differentiators of IRQA are the following:

- **Process – focused**: IRQA has been recognized by several international analysts as the best tools for supporting Requirements Management and Definition processes. IRQA provides visual block diagrams that allow the users to easily understand the process, and navigate through it, adaptable workflows that allows to define possible transitions of states and related actions, traceability model definition, and many other supporting capabilities.
- **Enterprise-wide model**: IRQA provides an unparallel concurrent access model, allowing the users to work in parallel on requirements of the same type, establishing access at element level (even more detailed than the document level) and controlling the access and/or visibility of the requirements data and attributes, allowing all the organization to access to the repository while

protecting the information. On the other hand, IRQA offers a best-of-the-market distributed work model, allowing the requirements interchange between IRQA repositories or even with other RDM solutions and *repositories* that your providers could have. This approach, combined with a fantastic *offline* work support that allows roundtrip information export/import, detecting the changes introduced in the repository, makes IRQA a really Enterprise-wide solution.

- **Flexibility**: IRQA doesn't impose any way of work or process. IRQA has been used for many years both in the Systems Engineering and Software Engineering markets, which shows how flexible the solution is in order to support any process defined in the organization.
- **Workflow Management**: IRQA provides workflow support when changing Requirement and any other element's state, giving the ability to execute open and easily-programmable tasks defined by easy-to-use Visual Basic scripts, most of them already developed and available for free. Moreover, IRQA allows defining a workflow in terms of Element Types dependencies and relations, enforcing also to use a specific traceability model.
- **Special attention to reusability**: IRQA has been designed with reusability in mind. Creating reusable components that could include not just requirements, but also the related tests or services specifications along with the existing traces between the artefacts is as easy as selecting the artefacts that will be part of the component, allowing to share the definition, specification and verification of the component reused. Reusable components can then be published in order to be reused by other projects in three different ways: copy (a copy with no reference to the original component), reference (a read-only reusability of the component content that could be updated when the original component will change), and copy by reference, a mix of the previous options, that allows the new project to reuse a component in an independent way but always being able to upgrade to the new version. Components can additionally be reused fully or partially, providing a better approach to the real-world.
- **Process Assembly:** Due to the fact that IRQA provides an integral support to all the Requirements Management and Definition activities and stores in a unique repository all the development artefacts, it is possible to assembly a process that covers all the development lifecycle. Process definition in IRQA is not only related to Requirements, but also to tests, models and change requests among other.
- **Customization:** IRQA is fully customizable, using graphical interfaces and through very intuitive views. Customizations can then be exported in the form of project templates, and reused in several projects, enforcing a common process and user interface. Everything in IRQA, even the language and texts of the solution, can be customized. IRQA allows customization of users, groups, security model, access model, requirement types and attributes, tests, services, user scenarios, workflows, problem domain models and traceability models among other, while

still being able to integrate with third party tools and being possible to map user defined attributes between both solutions.



Business Requirements and Solutions Repository

IRQA is repository oriented. Its repositories and projects are created on top of a commercial database (ORACLE, SQLServer or MSAccess). Requirements, solutions and test cases are stored in that database in a transparent way to the user. Within the repository the administrator will define the IRQA users (that may be created directly into IRQA or imported from one User Management tool like LDAP or Active Directory) that may be assigned to the different projects stored into the repository. The Project is the basic work unit of IRQA where we can find all the requirement specification info. The entire set of elements used in the specification (like requirements, use cases, tests, etc.) will be found into one of this project. Additionally, IRQA will allow users to reuse the elements from one project to other project in the repository.

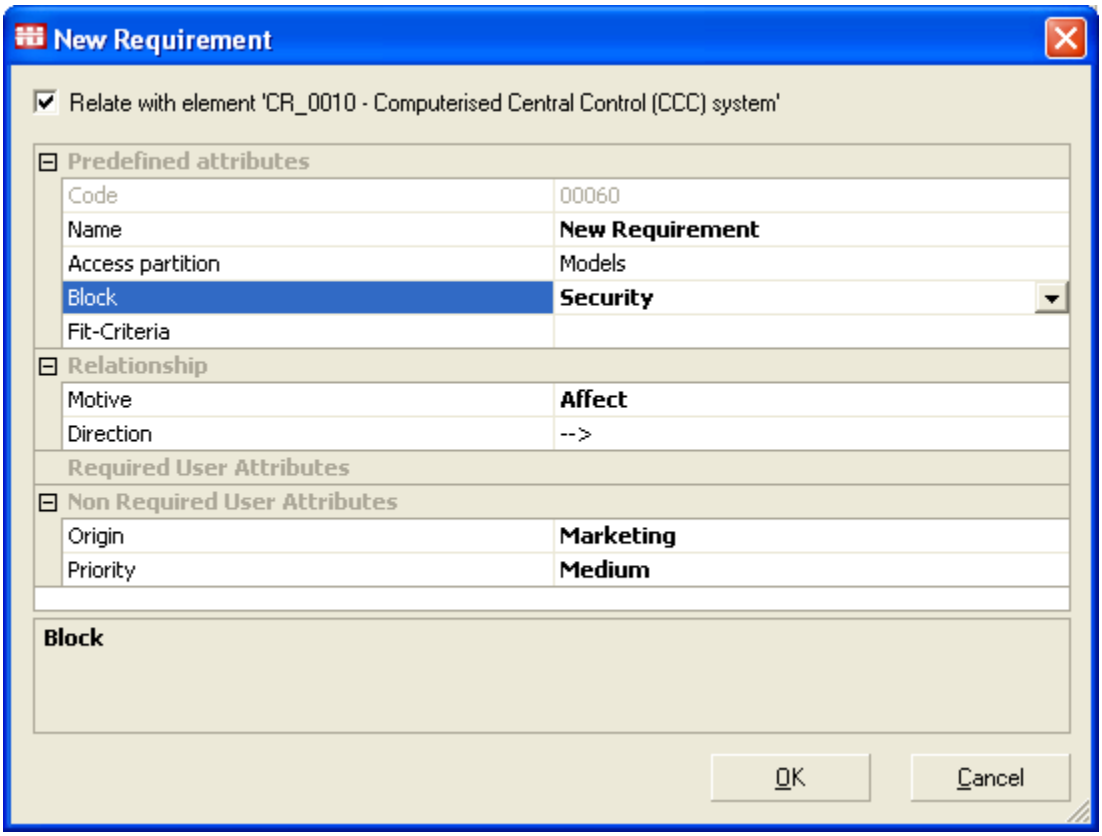
One important characteristic of IRQA that must be understood is that **it is not document-oriented, but information-oriented**. That means that the single unit of work is the requirement, not a document. However, as we will see, it is possible to import/export to documents easily and moreover, the requirements can be managed internally to the tool using a document-view.
Document views are one of the different points of view of the information, in which it can be seen and organized, but IRQA provides many other in order to support from simple to very complex requirements structures.

New elements and changes over the existing ones are immediately saved in the repository, and made available for all the users, allowing them to collaborate in a more information-centric approach, instead of a document approach.

Capturing Requirements/Identification

IRQA supports both, manual and automatic input of Requirements, Services/Solutions and Test Cases.

1. First of all, the manual capture of requirements and other elements allows the user to manually input requirements into IRQA. This manual capture is very simple and intuitive and helps to the creation of different requirements types. As IRQA allows to define different requirement types (like functional, non-functional, different types of non-functional, etc.) it is possible to specify upon creation the type of the element that is going to be created, which means that the tool automatically will restrict the traceability for this element (based on the traceability policies defined by the user) and also will allow the users to fulfill the attributes for that type (as IRQA allows to define the scope of attributes based on the requirements types). See requirement creation dialog below:



Automatic capture of requirements, services and test cases from external documents is also supported.

IRQA provides an Import tool for this process. This tool allows the importation of requirements from external formats. Requirements can come from MS Word, MS Excel or XRI (an XML format designed by Visure Solutions for Requirements Interchange between RM tools).
If data to be migrated are in MS Word or Excel files, these files will have to be reviewed in order to check if IRQA Import tool will be able to import them as they are. Normally, some kind of customization needs to be performed on the files (like defining styles in Word files, standardizing column headings in Excel files, etc.)
If data to be migrated are in RM tools, XRI files containing this data will have to be generated. Visure has developed compatibility with some of the most common RM tools.

More information on the capture process:

- ***Setting up the capture criteria***

Depending on the file being captured, it is needed to set up the capture criteria:

- *Requirements import from MS Word files*

IRQA Import Tool can capture requirements contained in .doc files in text format. The following information can be imported from this type of file:

- Code, name, and description of requirements.
- Hierarchical relationships between requirements.
- Attribute values.

None of these fields are required in the file. If no capture criteria are entered for any of them, the tool will do the following:

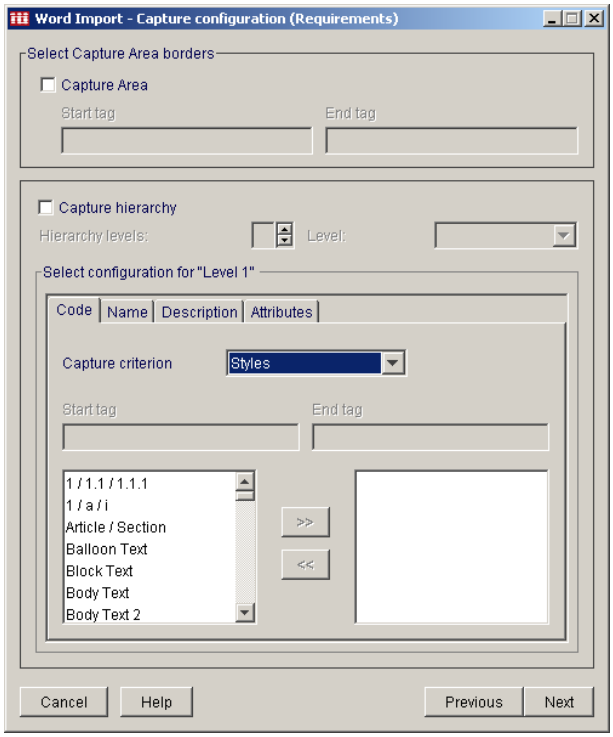
- Code: The code criteria can be set later, or the criteria defined in the IRQA project by the user doing the import can be used.
- Name: The tool will assign the character "-"for the requirement name.
- Description: This field will be left blank.
- Hierarchy: No hierarchical relationship will be established between the imported requirements.
- Attributes: The default values specified in the project will be assigned.

However, either the code or the requirement name is required in the import, since this is the criteria used to determine the start of the capture of a new requirement.
It is also important to note that it is possible to read a code from the file as a criterion to know when to start the capture of a new requirement, and then assign a different code to the requirements read from the file.
The file should have the following characteristics in order to be imported correctly:

- ***Style-based capture***

The simplest criteria for capturing requirements from text in a Word document are the style-based format.

Each one of the types of information to be captured (code, name, description, attributes) must be in a different style. When the tool encounters a paragraph with the style specified for a field type, it interprets the content of the paragraph as the value of the field.



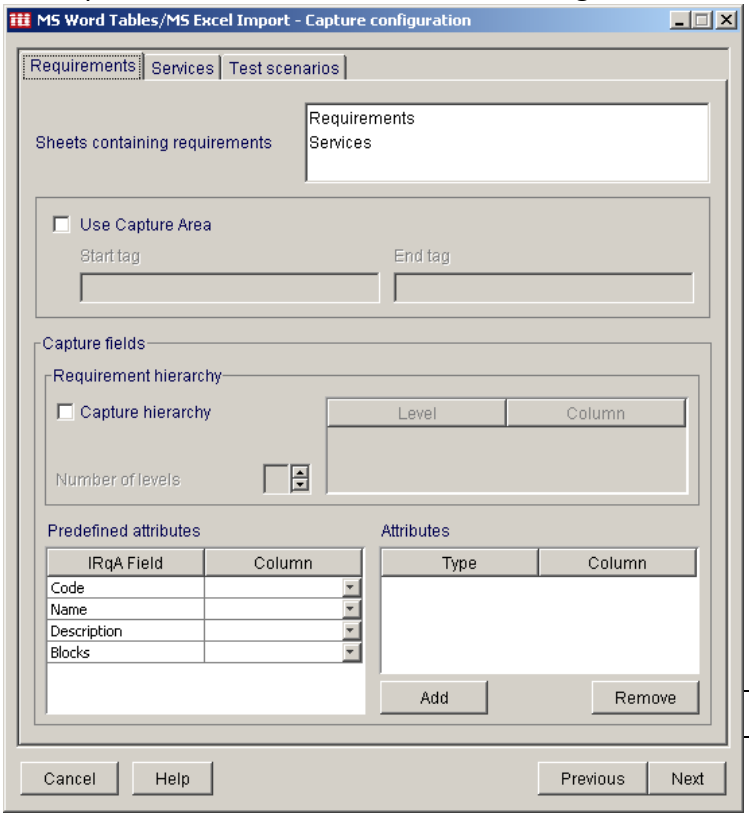
It is possible to use multiple styles for a single field type (for example, the description could be in the styles Normal and Body Text).

- ***Tag-based capture***

Another criterion available for capturing requirement information from Word text documents is the delimitation of fields using tags.

These tags will be character strings (for example <START_CODE>, <END_CODE>) that must appear as isolated paragraphs in the document, with a carriage return at the end of each.

Both capture criteria can be combined; for example, the requirement code could be identified by a specific style, and the descriptions delimited with start and end tags.



Requirements import from MS Excel files

Each worksheet in the Excel file is analyzed and: each requirement appears on a row and the fields associated with it (code, name, description, blocks, attributes) in the different columns. The field to which each column corresponds is identified by the name that appears in the column heading.

The first row with text in the Excel sheet is interpreted as

the heading of the table to be imported.

The following information can be captured:

- Code, name and description of requirements.
- Hierarchical relationships between requirements.
- Enumerated multi-valued attributes.
- Attribute values.
- Association of requirements to blocks.
- None of these fields are required in the file.

Each requirement appears on a line and the fields associated with it, in the different columns.

The fields to be captured are identified based only on the column of the table in which they appear. The field to which each column corresponds is identified by the name that appears in the column heading.

Any name can be used for the names of the columns in which the code, name, description and association to blocks appear.

In the columns with attributes, the name of the attribute in the IRQA project must appear. The read values must be valid according to the type in the IRQA project. Otherwise, the read values will be ignored.

Requirements import from XRI files

There is no need to setup up the capture for XRI files as all the information needed is contained within the file and the XRI format. Anyway, through XRI it is possible to capture only requirements (nor test, nor services) and the relationships between them.

Detecting changes in captured requirements. Change management mechanism

Once the file to be captured has been read, the tool analyzes the requirements read from it and compares them with the existing requirements in order to detect the differences, if any of the captured requirements already exist in the project.

The criteria to decide whether a requirement is the same, is that the code assigned to the captured requirement is the same as the one that it has in the IRQA project. Also remember that in the import process, it is possible in some cases to read a code from the file and then assign a code different from the one that was read. In this case, the code that is used for comparison is the assigned one, and not the one read originally from the file.

The tool does not analyze possible changes in requirements that are checked-out by a user other than the one who is importing them. This is due to the fact that these requirements will not be able to be modified in the import process, just as they would not be able to be modified manually (the only exception to this rule is if the importing user is the project administrator, who always has permission to modify, check-in, or check-out requirements, regardless of their status).

Selection of requirements to insert into the project

The user can select the requirements to be actually inserted into the project from among the requirements read from the file.

Keep in mind that it is only possible to select requirements that fulfill the following conditions:

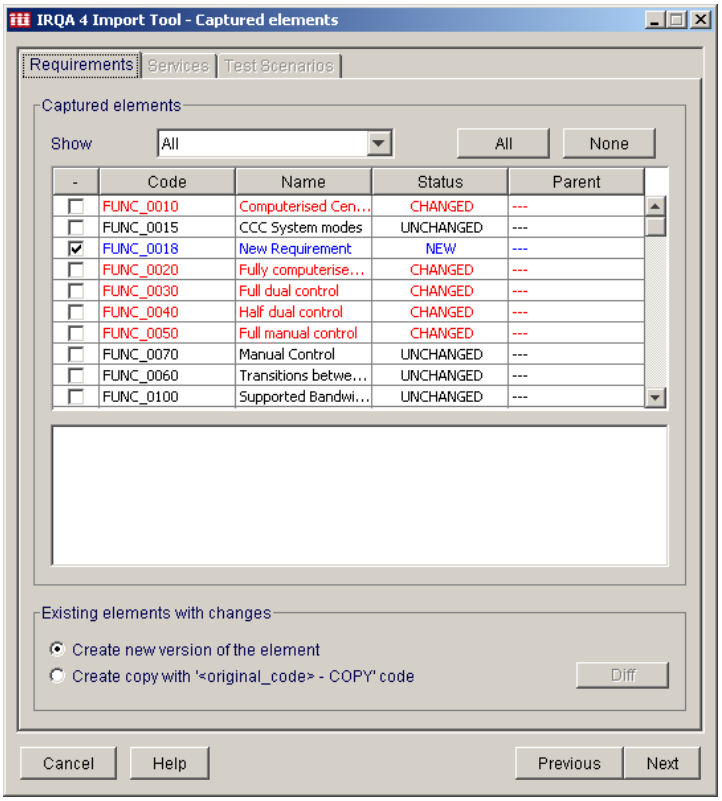
- New: The code assigned to the requirement does not coincide with any of the existing codes in the IRQA project.

- Modified: The code already exists in the IRQA project and the tool has detected a change. Note that this also implies that the requirement is checked-in, or checked-out by the user who is carrying out the import.

In the case of each of the modified requirements, the user must select how they will be incorporated into the project:

- As a new version of the existing requirement.

- As a new requirement, with a code that is the same as the requirement that already exists in the project, adding the string "-COPY" at the end. In this case, a direct relation will also be created between both requirements with the "copy of" motive.



Insertion of captured requirements

Once the file of .doc, .xls or .xri type is read, the process continues with insertion in the IRQA project of the information captured, taking into account the configuration criteria previously input.

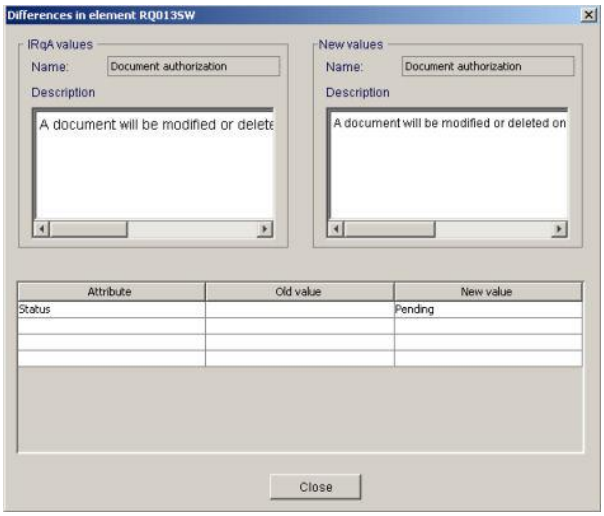
Selection of requirements to insert

The user may choose from the requirements that have been read from the file, those he wishes to insert in the project. To do so, the following dialog is shown:

By default, all the requirements read in the file are shown. The fold-down menu displayed on the head of the list may be used to select the elements the user desires to show. The available options are:

When a requirement on the list is chosen, the description read from the file for that requirement appears in the bottom window (it only shows the text parts, neither tables

nor images are shown, although both are inserted as part of the description of the requirement in the IRQA project).

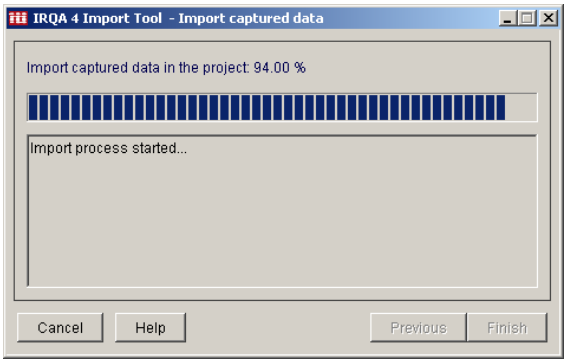


The user may select the requirements to include in the project from the list shown. To do so, these must be marked in the column on the left of the list. All may be marked, or all the marks may be eliminated using the options provided for that purpose on top of the list.

For requirements existing with changes, the differences between the read version and the one existing in the IRQA project can be shown selecting the requirement and clicking on the *Diff* button.

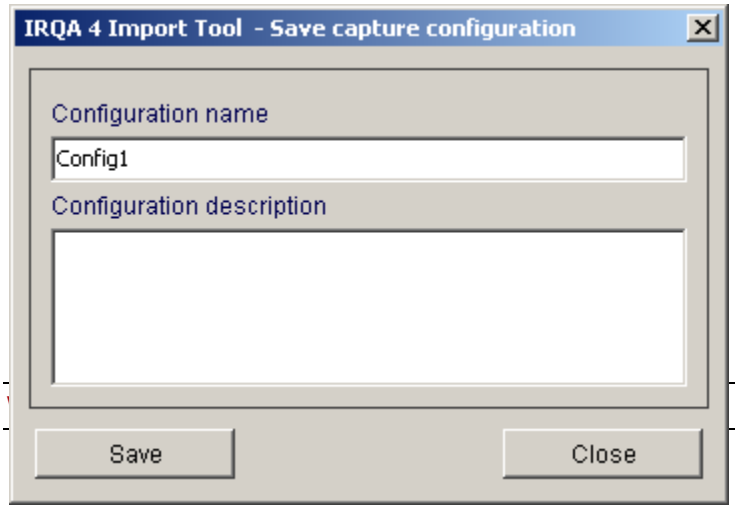
Insertion of elements in the project

Once these options are chosen, the "Insert" button is clicked which sets off the process of insertion of the elements in the IRQA project.



Creation of configuration for capture

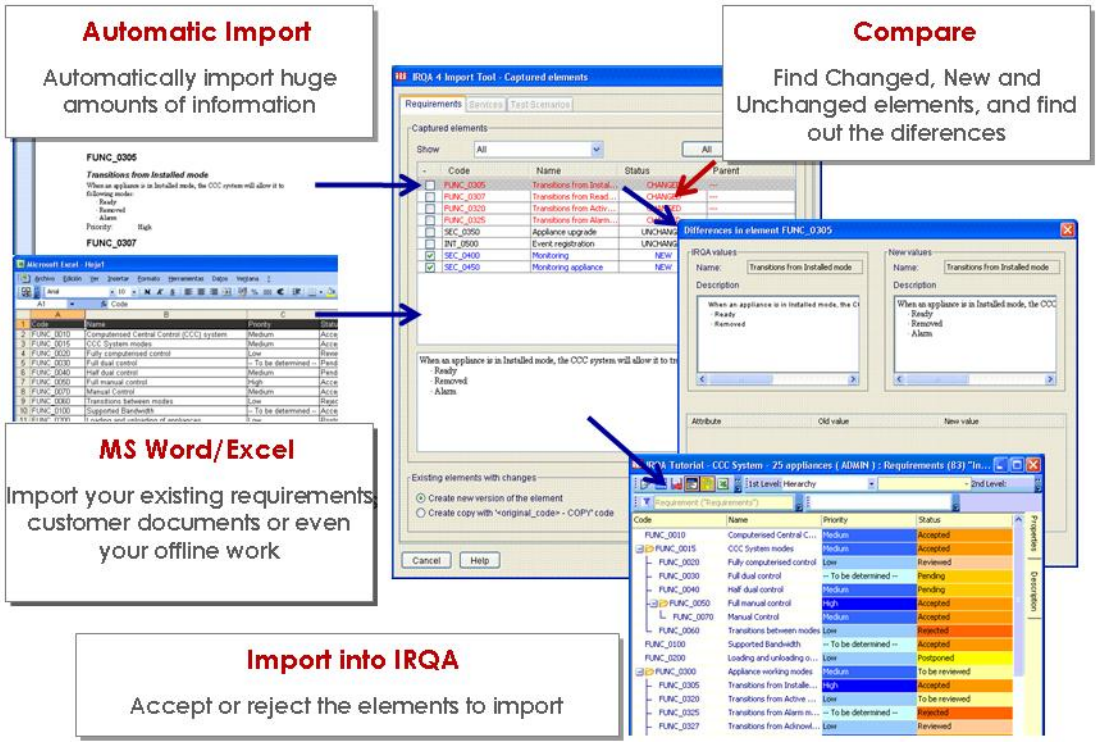
If a Word or Excel file has been imported, when the button "End" is clicked after inserting the requirements in the project, the following dialog box is displayed:



The user is allowed to save the configuration just defined for the file capture, giving it a name and description and

clicking the button “Save”. Thus, the configuration will be available for future automatic captures. The capture configurations for Word tables and Excel files can be used interchangeably for both file types.

If you do not wish to save the configuration, click directly on the button “Close”.



Additionally, the import tool has an unattended version that can be launched as a batch process and that will import automatically the requirements and put them into IRQA based on the configuration files used.

This capability is currently being used to synchronize the IRQA database with information from offline third party applications, and also to automate the exchange process with customers and suppliers.

Finally, apart from this Import tool there is a plug-in for MSWord and MSOutlook that helps to import requirements into IRQA by an interactive mean, mouse highlight.

IRQA can either import the unique IDs for Requirements, Services and Test Cases from the document or assign them following the numbering pattern specified by the user (Prefix, Suffix, number of digits, etc).

During the import process, elements imported from documents can be assigned to different subsystems, access partitions or specific element type inside the IRQA repository.

Capturing System Elements Structure

One of the major strength of IRQA is its ability to represent (both textually and graphically) the system/domain of the problem being modeled in the tool and also the element metamodel used. For these purposes IRQA comes with several diagrams like:

- **User definable Block Diagrams**: Blocks are elements of the same type (requirements, uses cases, tests, concepts, etc.) that share some common characteristics (for example, functional requirements, customer requests, non functional security requirements, etc.). These blocks are used as types and subtypes into IRQA. Apart from being element containers blocks may be represented graphically using a block diagram. These diagrams are used to represent the requirements process used in the project, below there are several examples of a block diagrams configured for different sectors:

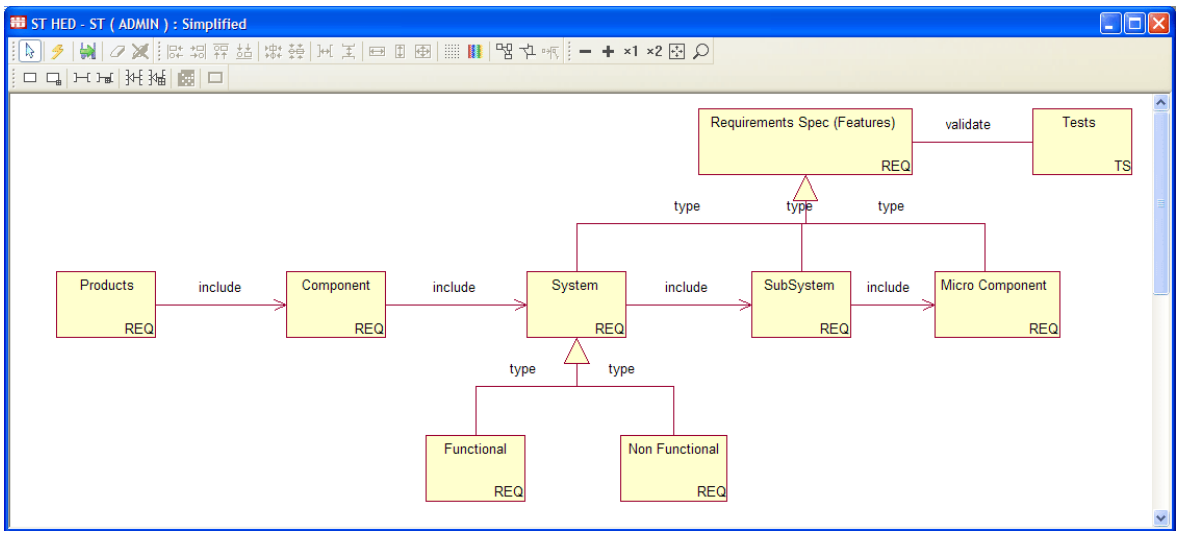


Figure 1. Typical representation of microelectronic development

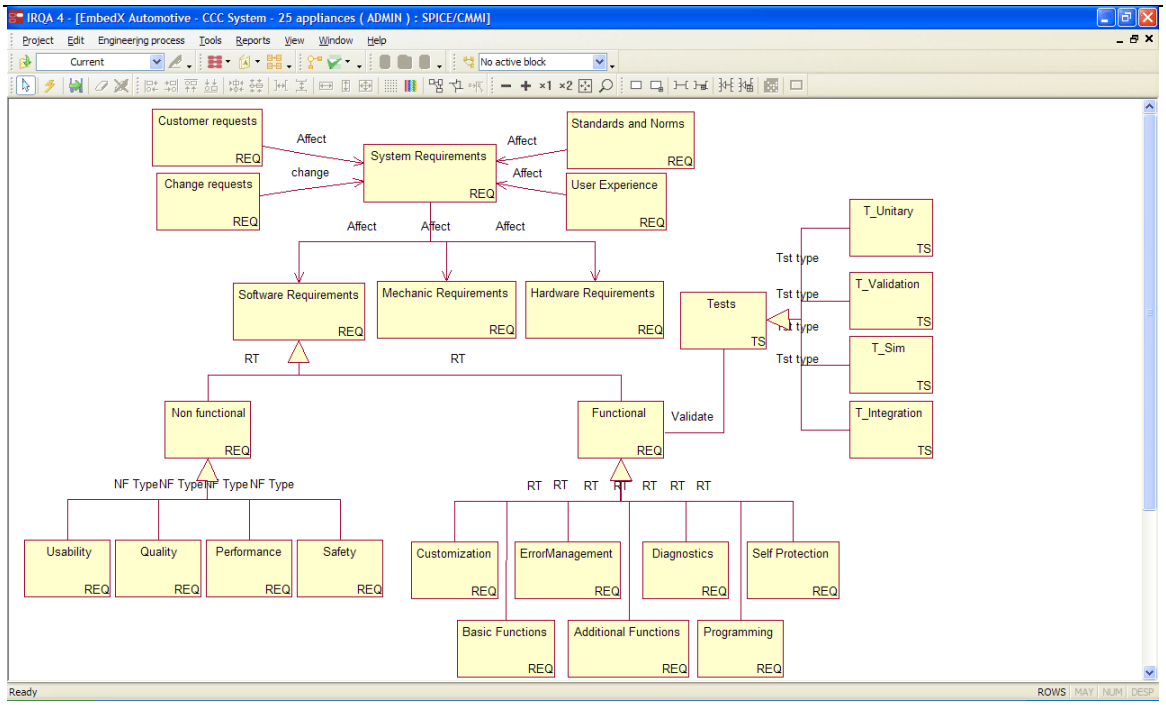


Figure 2. Typical representation in the automotive sector (CMMI and SPICE compliant)

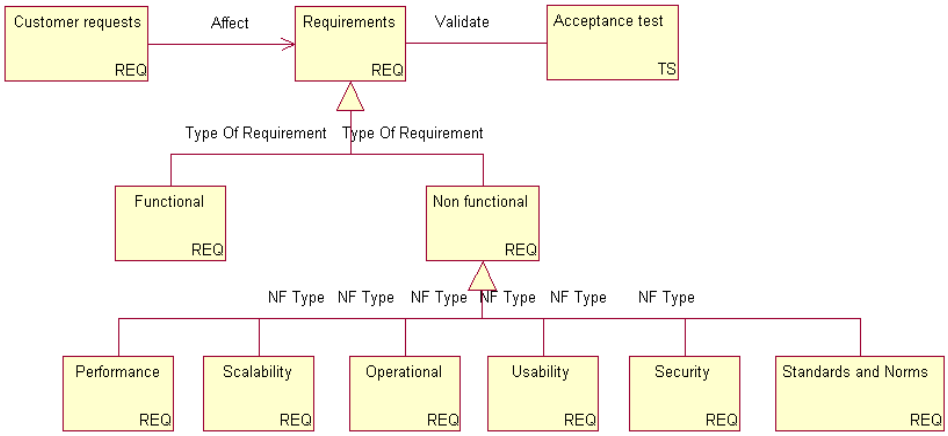
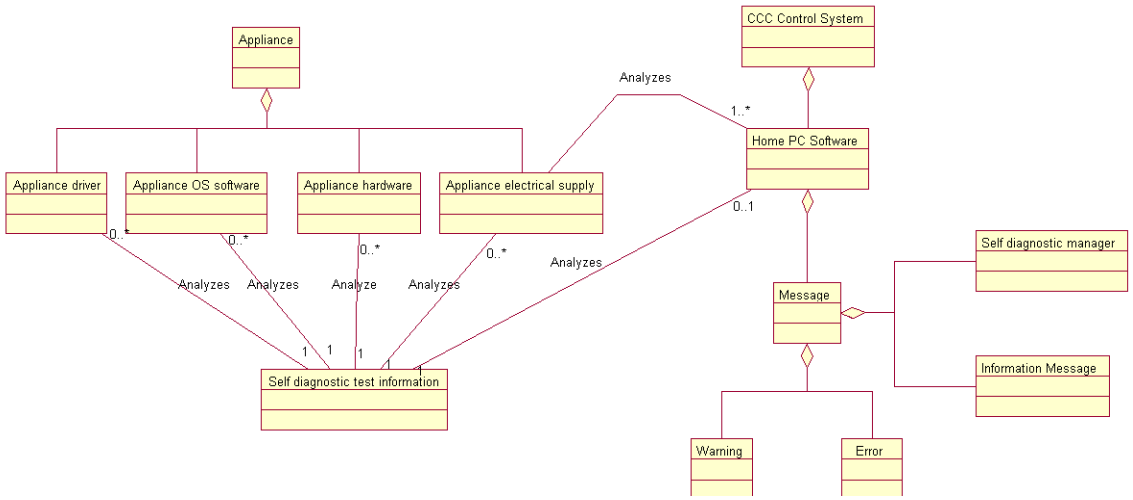


Figure 3. Typical representation for software development

It is possible to enforce the users to follow the diagram as it is possible to restrict the available relationships to the ones represented in the diagrams, for example, in this case we can trace all the elements that are business domains with requirements, but we cannot trace then directly with tests or with design constraint. This restriction will help us to avoid errors and will make the process easy to follow to the final users.

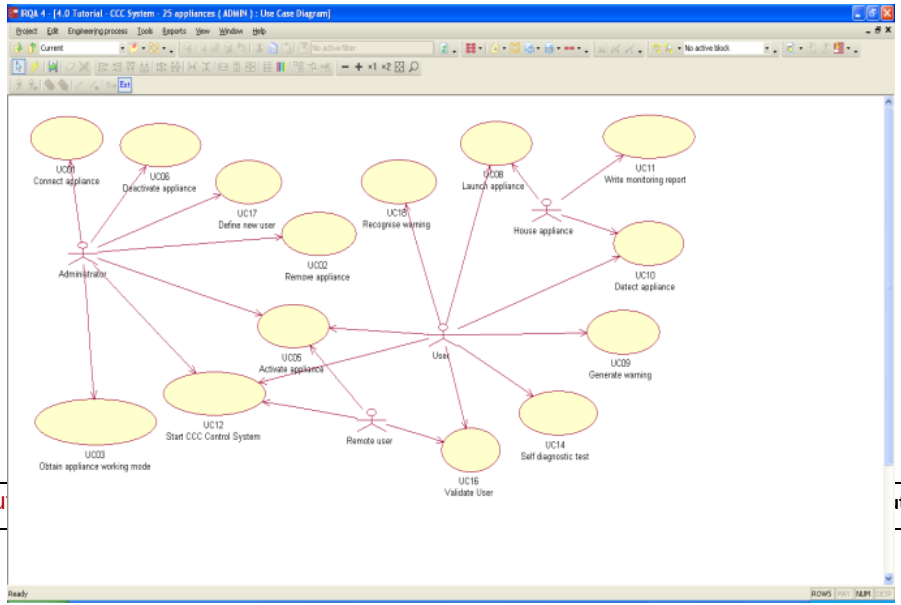
The elements represented in the diagrams are also automatically created as textual elements that may be traced with requirements or with other specification elements.
Block diagrams may serve as Traceability Diagrams, WBS or functional decomposition, among other.

- **Concept Diagrams**: Concept diagrams use the UML class diagram notation for representing graphically the system being modeled or specified in the project.



These diagrams are the graphical representation of the textual concepts. Any element that is created in the diagram will have a corresponding textual concept. These concepts may be later traced with requirements. Not only the concepts but also the attributes of the concepts and the link between concepts may be traced with requirements, helping in this way to identify the source of the requirement and the part of the system/domain that is affected by the requirement.

- **UML Diagrams:** for example class diagrams, state diagrams, use case diagrams,



flow/sequence diagrams, etc. Those diagrams can be created inside the tool, traced to different elements and can also be exported as images (BMP). Additionally there are special views to work with the elements stored in those diagrams, for example, the use cases from a use case diagram:

- **Domain diagrams:** They are used to represent Domains and their relationships (basically used as Architecture diagrams or graphical representations of subsystem decomposition).
- **Others:** DFDs and Context diagrams.

Folder Structure

For the classification of elements IRQA uses the blocks (previously described). Administrators may create as many blocks as needed in order to contain the requirements (or other elements) that are going to be created in the specification. Take into account that the block architecture of IRQA is really flexible as an element can belong to several blocks at the same type or be moved from one block to another.

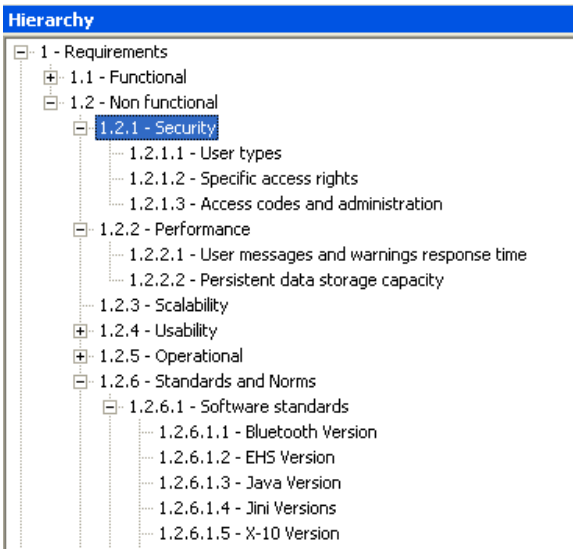
This is extremely useful and powerful when designing complex system. It allows, for instance, to define a structure e.g. functional, non-functional, usability, escalability, etc. and apply that structure to other structures e.g. User requirements, System requirements, component requirements, or even, component X requirements, Component Y requirements.
So, for a given element, it is possible to define it as a performance requirement, while being, at the same time User requirement from Component X.
This is useful when working with all performance requirements of all the components, for example.

Folder structure provides a single dimension, where as the blocks in IRQA provide a multidimensional organization of elements, which can be defined graphically through the block diagrams described above.

At any given moment, this structure can be represented in a Folder like, or Tree like View in the Document View.

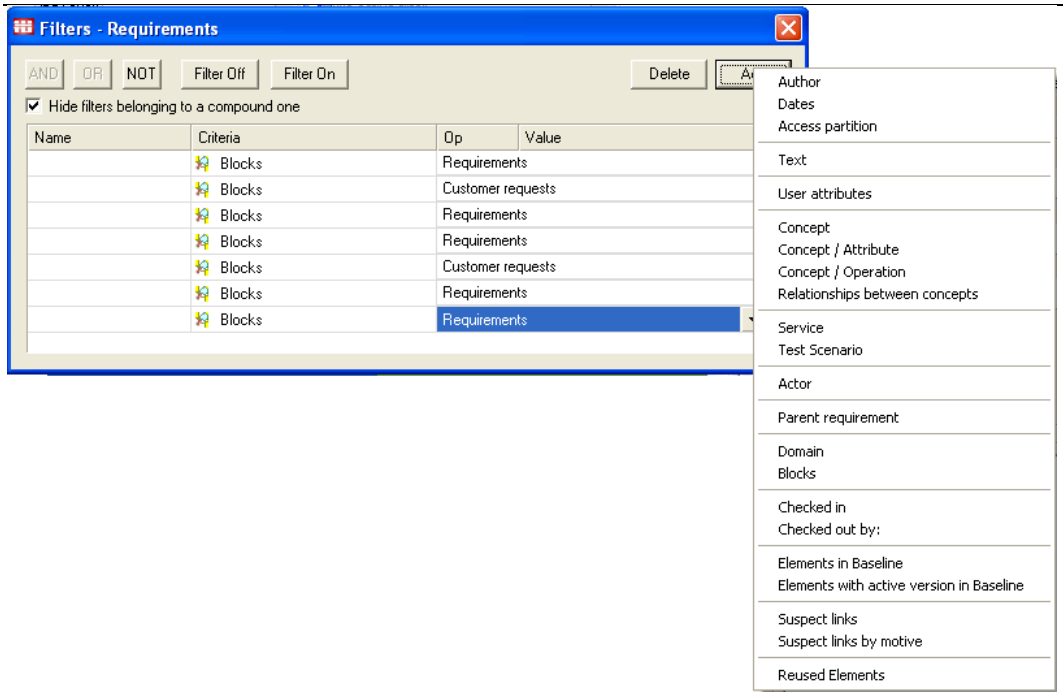
Additionally blocks may be used to define the attributes scope and traceability policies of the elements. Finally, blocks may be represented textually in the document view

which is basically a document representation of the block diagram where the blocks are represented as chapters or folder that contain the elements:



Search and Retrieval

Searching may be accomplished using the powerful filtering capabilities of IRQA. Using filters, users may look for almost everything in the requirements; who created the requirements? When? Values of a given attribute, text in the name or in the description, elements related with a given element, elements with a given parent, etc. these filters can also be mixed using logical operator (AND, OR, NOT) so we can search for a requirement created by John Smith AND created on July the 5th AND in status pending (for example). See below for an example list of filters:



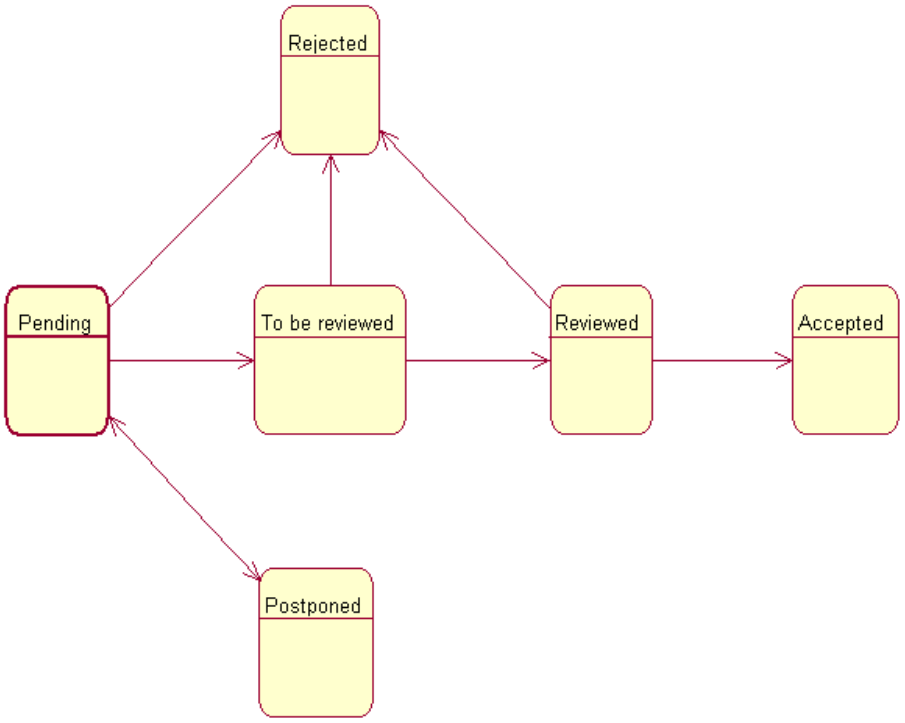
Classification and Indexing

IRQA works with a commercial database and everything is stored in the database.

IRQA has a set of predefined fields into each element like author, creation date, version number, etc. that are filled automatically. Additionally it is possible to define as many attributes as needed and also define them with the desired type (integer, enumerated, real, Boolean, etc.). These attributes, once created, may be applied to different elements types or blocks, for example, functional requirements may have a set of attributes and security requirements a different one.

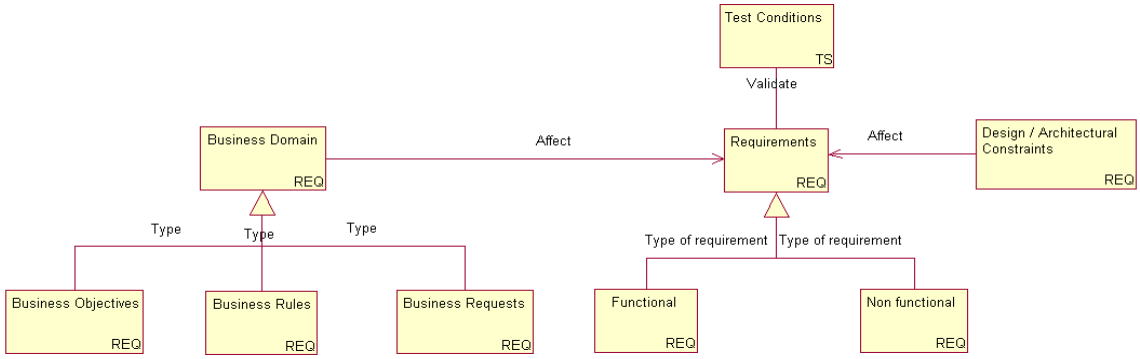
Through the plugins capability, the attributes can be automatically filled in based on any given criteria. For instance, a cost attribute may be calculated based on the cost of associated requirements. The status of a high level requirement based on the status of its related elements, and so on.

There is a set of predefined types into IRQA for the attributes (Boolean, enumerated, real, text, integer) but users may defined new types, for example, a type for percentage (from 0 to 100 and integer), a type for status (draft, under review, under approval, released) or even calculated attributes. Additionally, for enumerated attributes it is possible to define a workflow associated to the attribute. With the workflow it is possible to define the available transitions between the different values and also which user has the rights to perform each transition.



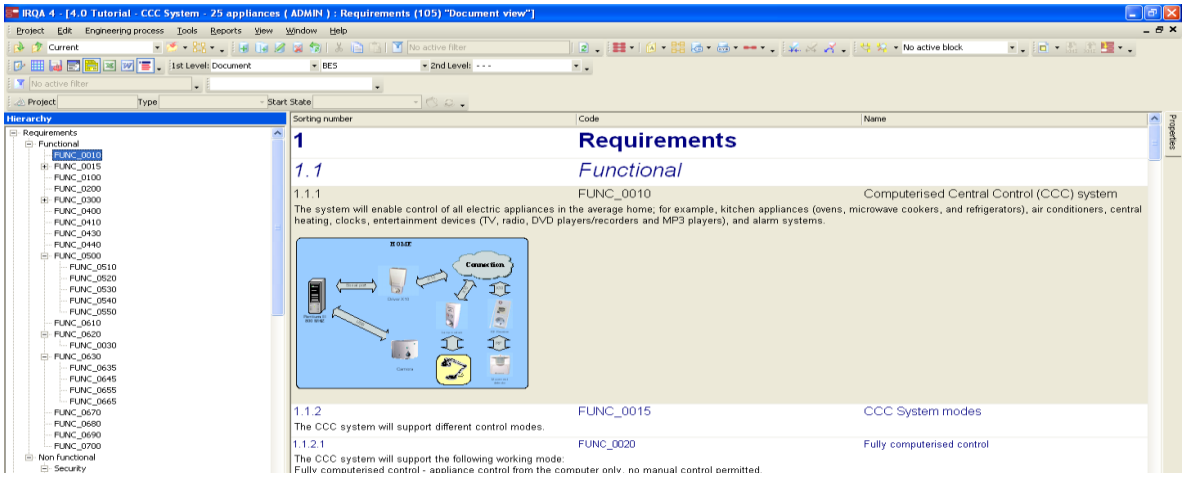
Finally, it is possible to associate VB Scripts (most common ones already implemented and available for free) to the transitions so they will be executed every time an element switches from one value to another one. For example, it is possible to define a script that sends an email to all the analysts every time a requirement is approved. Using these scripts it is possible to automatism some tasks

As told before, IRQA is characterized by its flexibility. Any type of requirement can be captured into IRQA and then categorized as needed. For example, it is possible to create a block diagram that represents our requirement process following the different types enumerated above:



As an exmample, this diagram represents an easy process in which our requirements (that can divided into Functional and non-functional) are affected by the Business Domain (that can be Business objectives, Business Rules and Business Requests) and also by the Design constraint. Finally, our requirements (functional or non-functional) can be validated through Test Conditions.

It is even possible to enforce the users to follow the diagram as it is possible to restrict the available relationships to the ones represented in the diagrams, for example, in this case we can trace all the elements that are business domains with requirements, but we cannot trace them directly with tests or with design constraint. This restriction will help us to avoid errors and will make the process easy to follow to the final users.
The classification of different types of elements is carried out by *Blocks* (defined early in this document). In IRQA users may assign different attributes to elements depending on their belonging to blocks, for example, you can assign a priority attribute to the elements contained in the block Business Domain and a status attribute that will apply to the elements in the block Requirements.
This block classification can also be used in a more textual way (instead of the graphical one) in what we call the document view.
In this view the blocks are shown as chapter/subchapters and the requirement belonging to any block are shown under the corresponding chapter:



Check-In/Check-Out

IRQA provides an excellent configuration management.
Configuration Management of the requirements is performed at requirements level, and not at document level, because of the well known concurrency, access and collaboration problem the document approach has proved to have.

Requirements are independent entities that can be assigned to different documents and classifications.

With this in mind, in IRQA we can check who is working with which requirements, not only with the whole document but with given requirements. This is a extremely powerful feature as the blocking of elements is performed at element level, if one user needs to work with a given requirement only that requirement is blocked for other users, not the entire document or a set of requirements.

Nevertheless, a complete set of requirements or elements can be blocked en-bloc, for a specific user to work with.

IRQA works with a check-in/out system similar to the one used by SCM tools. Elements are checked-out to work with them and once a user finishes they are checked-in again. In the process a new version of the element is automatically created.

Checking-out an element only makes that element accessible in write mode to the user without downloading any information to the local machine. In this way, a user checking out an element in the office, might be able to continue working with that element at another location without any problem or restriction. Whenever the changes are performed, the user can then check in the element.
This means that the user can undertake a set of changes without the restriction to save them (and create new version) whenever closing the document or PC.

Finally, there is a Discussion forum in all elements that can be accessed by each user with read rights in the element and can be used to add comments, discussion, reviews to the requirements, etc.

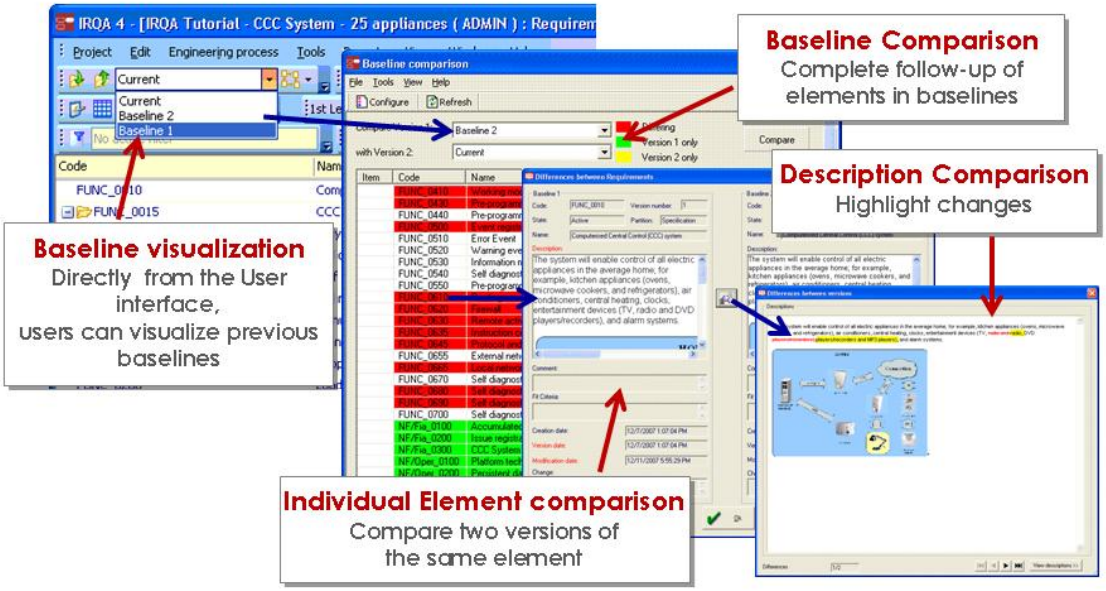
Version Control

Versioning is controlled automatically by IRQA as explained in the previous section (see above). At any given moment users with the corresponding rights may check the differences between 2 given versions of individual elements or compare them.

The version control of sets of elements (documents) is performed through the partial (or complete) baselining capability in IRQA, which allows to compare, as for the individual elements, all the information contained in that set, including traceability.

Baseline management and individual versions management

Elements can be tracked at individual level or at project level through versions and baselines, achieving a total control of the history of the project through detailed comparisons



Auditing

As every change needs to be performed by checking out the element, everything is stored and we can check at any given moment "who did what"into the tool
Complete individual requirements (and also other elements like services, test scenarios, diagrams, blocks...) evolution through version control may be maintained in the repository.

The history of each element may be tracked through the versions and the rationale of each change may be saved.

Requirement date, time, user, change reason, among others are recorded in addition to requirement name and description changes in requirement history.

Users can compare two different versions of an element obtaining the differences between them.

IRQA also provides a logging capability at different levels that stores, depending on the selected mode relevant information on the activity of IRQA.

Integration with Desktop Applications & Interfaces with other Tools

IRQA is characterized for being an open tool as it integrates with several application and offers different ways to integrate with other tools (like an open API or plug-in support).

This is a list of some of the current IRQA integrations:

- Modeling and design tools
  - Enterprise Architect
- Tests tools
  - HP Quality Center
- User configuration tools
  - LDAP
  - Active Directory
  - Windows Domain
- SCM tools
  - SourceSafe
  - Subversion
  - Any compatible with the SCC protocol
- Semantic analysis tools
  - IRQA Quality Analyzer
- Issue tracking
  - MKS Integrity Manager
  - JIRA

Additionally to these integrations, IRQA offers an open API for COM languages (C, C++, C#, .net, VB, etc.) and Java. This API is used by Visure itself to program integration or simply to program any software that needs to access and manipulate the data within IRQA. This API interacts with the IRQA database and even it is possible to get data directly from the database with different queries it is recommended to use the API to access to the data as it warranty security and even if the DB changes, the programs will be still valid.

Apart from the integrations and the API, IRQA offers a complete support to plug-ins. These plug-ins work as eclipse plug-ins, they are stored into a given folder and will be accessible from IRQA. These plug-ins may be used to program integrations or to expand the capabilities of the tools using a well known language like C/C++. With the help of the plug-ins new buttons, toolbars, tabs or options may be created into the tool.

IRQA also supports import/export from external formats like MSWord, MSExcel and XML. With this direct import and export it is really easy to create a requirements document just by clicking into one of the toolbars button which will create a word or excel document with all the requirements in the view. Additionally, these document may be also imported in the tool using the import tool (as described into the section 2) so these document may be used to distribute requirements, then they may be modified and then, we can import the changes back into IRQA. Especially useful may be the XML import/export, used to distribute requirements to different installations of IRQA

Finally, these imports/exports from external formats may be used to interchange requirements between different applications that support the importation of excel or word document (for example, MS Project).

Security

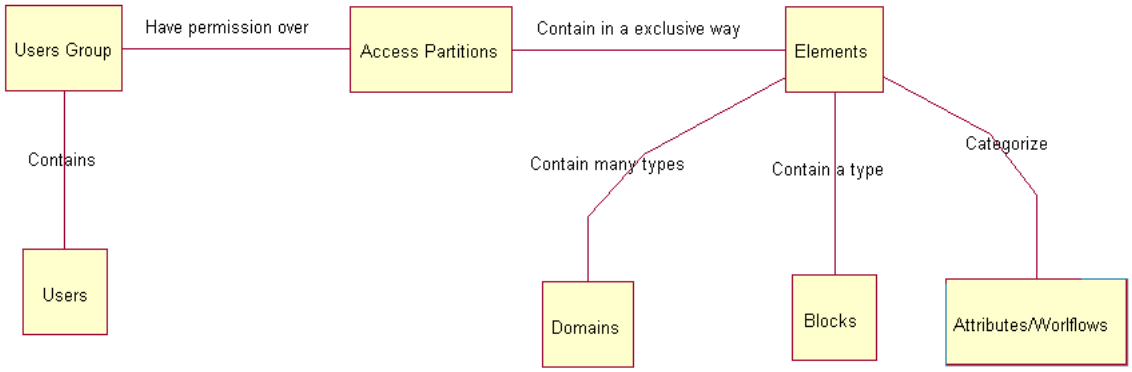
The rights management into IRQA is really detailed and Administrators can configure it to get exactly what they are looking for. Not only different rights (read, write and write own) may be given to different element types and subtypes (these subset of functional requirements have these rights, these other subset these other rights) but also different rights may be given to different parts of a requirement (for example: a given user may modify this requirements name and description but the status attribute is not editable for this user).

All this right information may be saved into templates and then, used to create new projects.

Approval workflows may be defined and even different rights may be assigned to the elements depending on the current requirements' status (for example)

Every single element within IRQA is contained into a partition. Basically partitions are containers of elements. Different read-write rights are assigned over those partitions instead of over elements themselves.
On the other hand, rights are given to user groups instead of over users directly. User Groups are basically the different roles users can take into a given project, e.g. Analyst, Business Analyst, Testers, etc. Users are assigned to User Groups. So basically, rights are assigned to Users Groups over partitions (Indirectly, to the elements contained on those partitions). This mechanism, despite seeming complex, it is really easy to use and really powerful, as it is possible to assign different rights over different elements (e.g. Testers may have read rights over functional requirements and read-write rights over tests) and even over different parts of an element, e.g. if we assign an attribute and the elements that take that attribute to different partitions we may give different rights overt the

attribute and the element, that is, we can have read-write rights over a requirement and only read rights over the attribute



When an element is created, it is assigned to a partition. This partition can be changed in future versions of the requirements, so we can modify the different rights easily during the element life. This can be useful, for example, to create a draft partition, where elements are created and only Analysts may have read and write rights over them. Once the element is ready to be shared to the rest of the users, we may switch it to a different partition where other users may have rights.

Retention and Disposals of Records – Records Security

IRQA accesses directly the information in the database. Some of this information may be stored in local cache for performance improvement, however, all this information (both stored in local cache, and travelling through the network) is conveniently encrypted and compressed both for security and performance reasons.

IRQA uses a check-in/out mechanism (described in a different section) that allows to check who performed changes over the elements ("*who did what*" and "*when*"), and this information is stored into the repository the desired period of time.

Additionally, IRQA allows putting into the requirements (or other elements) OLE objects that will be stored in the database along with the element. Using this functionality it is possible to store in the database documents or other information associated to the requirements.

Last, it is also possible to link external files to the requirements. These external files can be on a local folder system or even in a SCM tool using an integration based on the SCC

protocol. Using these two features it is possible to retain and dispose documents and records associated to the requirements.

Collaboration

IRQA has been especifically designed as a collaborative tool and it includes several features related with this subject.

- IRQA uses the standard check-in/check-out technique, so when a requirement is being modified, it cannot be modified by any other user until it is explicitly released, although other user can read it while the requirement is locked (if he has the appropriate read permission).

- Lock granularity is at the requirement level, allowing different users to work with different sets of requirements of the same type.

- With IRQA, users can set different partitions and the database will remain consistent among all "views". Furthermore, any change will be reflected dynamically across the database.

- For each requirement a Discussion tab exists, where every user with read permission can enter a text commenting on some aspect of the requirement, this will lead to a small discussion forum associated to the requirement

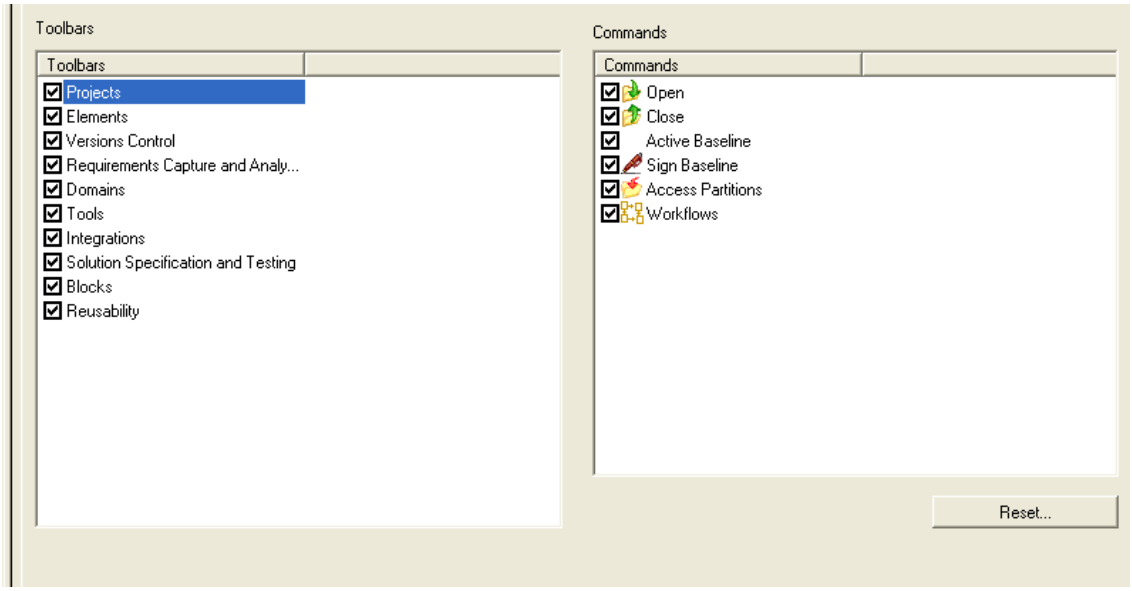
- As described before, several access rights may be defined over the elements (and even over different parts of elements) so the tool ensures that every user is able to perform the tasks he/she is intended to perform. This access rights can be changed (even automatically) during the life of the element or the requirements so we can given visibility over the element to different roles in different stages of the lifecycle of the element.

- As it is possible to assign different rights over different attributes and these attributes may be created as mandatory, it will be possible to force several users to fulfill their assigned attributes before creating the first version of an element.

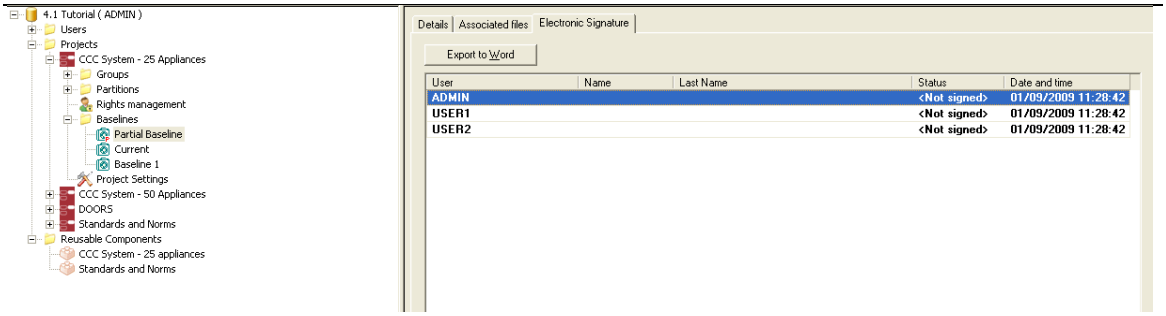
- Using the workflow system and the VB scripts that can be added to the transitions it will be possible to make certain check automatically (check that every attribute have been fulfilled upon changing the status)

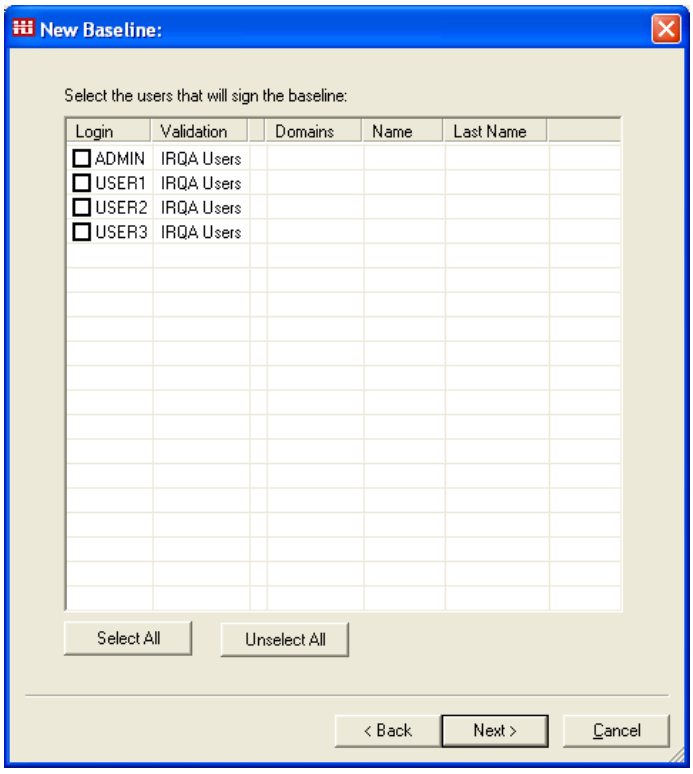
- There is a comment field associated to every requirement in the tool. This field is accessible to everyone with at least read rights over the element and can be used easily to create a discussion forum associated with the requirement where users may post question and these question may be answered by other users. It is even possible to use this field to create peer reviews

- Another of the great advantages offered by IRQA is the customization of the interface. It is possible to create a specific interface (including toolbars, views, columns, tabs, etc.) based on roles. Using this feature, every user may have his/her own options available, making it easier to know what they have to do (as only the needed options are available)



- Upon baseline creation it is possible to specify a set of users that need to check and sign electronically the baseline. Using this feature assures that the designed users have reviewed the specification.





- IRQA comes with a set of functionalities that helps the users to import/export requirements to external formats. Using the export capabilities is possible to create word or excel documents that will contain the desired set of requirements and attributes by just one click. These documents may be modified or reviewed in excel or word and then, the changes may be imported back to IRQA. Take into account that this import will be performed using a complete change management mechanism that will notify of the changes and will avoid any data loss. Using this requirement exchange mechanism make it easier to distribute the requirements to several users (even those without access to IRQA).

- Reusability

Recent studies have concluded that a large amount of requirements are reused from one project to others, that is, those requirements are exactly the same in several

different project. For example, 61% of requirements are reused in EADS/CASA projects. This usually happens with generic functionalities, style rules, standards and norms, laws, and so on. To avoid rewriting those requirements again and again or to copy/paste them IRQA provides a exclusive reusability system. Reusability allows to share/reuse sets of requirements from one project to others saving a lot of time as is no longer needed to rewrite the same requirements and is much more complete than just copy/paste.

The reusability unit is the component, which can contain requirements, services, tests, attributes and relationships between all of the elements in the component e.g. Performance or usability requirements, functionalities like user login, etc.

Components can be shared in different versions in different projects. If a newer version is published to a project, the project is notified in order to update (or not) to the latest version of the component. Using this mechanism users only need to update requirements in the original project and then, the change will be notified to all the target project, letting them to take the changes (or not)

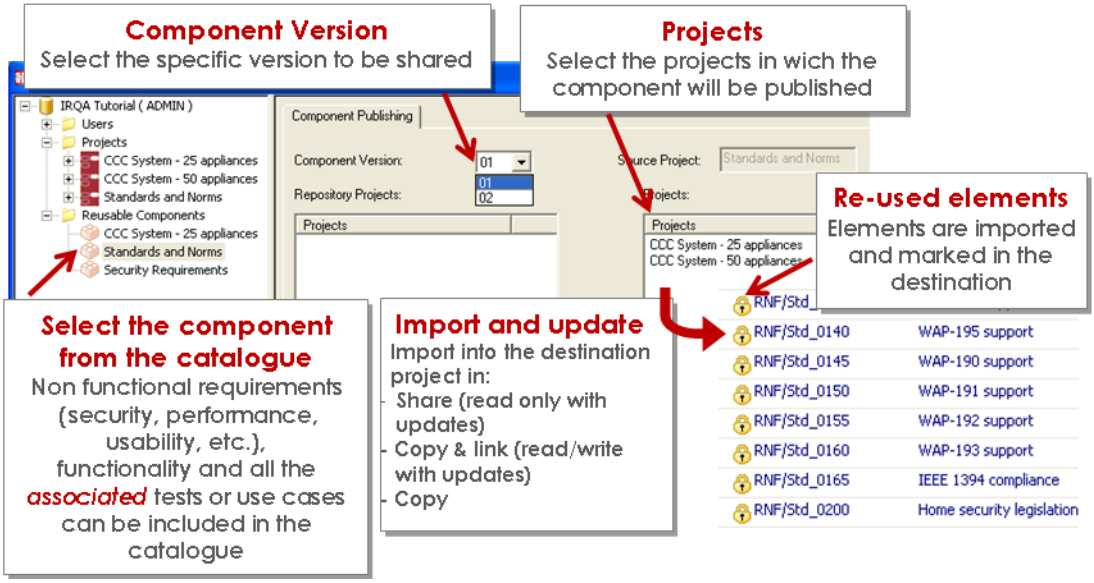
When a component is published to a project, it is possible to re-use it in several modes:

- Share: The elements are read only and cannot be modified. However, they can take attributes and can be related to other elements.
- Copy and link: The elements are imported in read/write, so they can be modified and updated. They can also take attributes and can be traced to other elements.
- Copy: The elements become part of the destination project and loose the link with the original one.

The same element (requirements, services or tests) can be included in several components in order to represent different “variants” e.g. basic product and advanced product.
This re-usability support can be used for product families and variants.

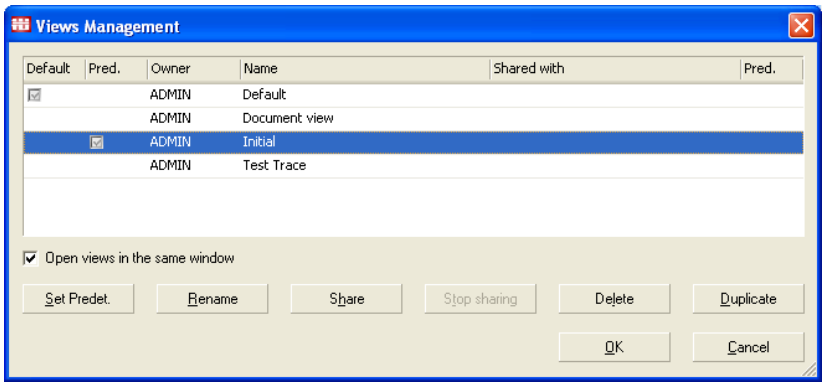
Save time and improve quality through the re-use of components

Requirements, Use Cases, Tests and the relationships between them can be "packaged" in components which can then be re-used through the whole company



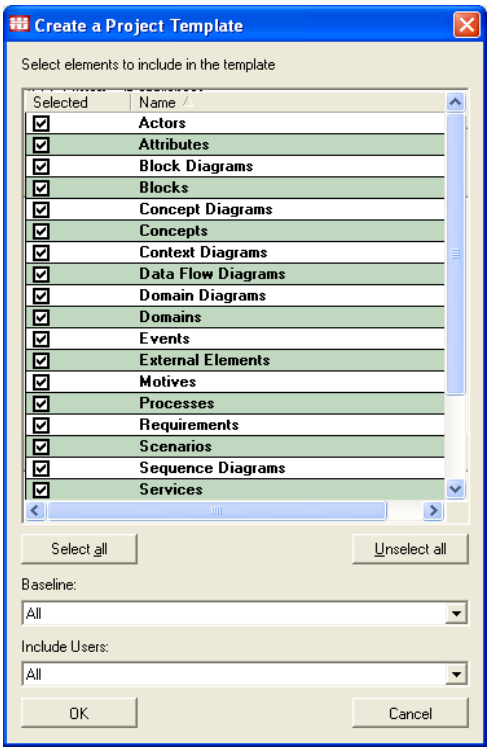
Views Configuration

Requirements can be displayed in several users views. Almost everything from the view can be configured by the user: Columns, Filters, Classification Criteria, Toolbars, etc. These views can be then stored and even shared to multiple users, so you can create specific view for the different user groups (roles).



Templates

Almost everything that has been seen defined can be saved in a template (From elements to views). That template can be used later to create new projects. Usually, all the business elements are stored into a template: Block Diagrams, Attributes, Views, Filters, etc. so every time we create a new project all this information is already defined and there is no need to create everything again, the project will be ready to start writing requirements.

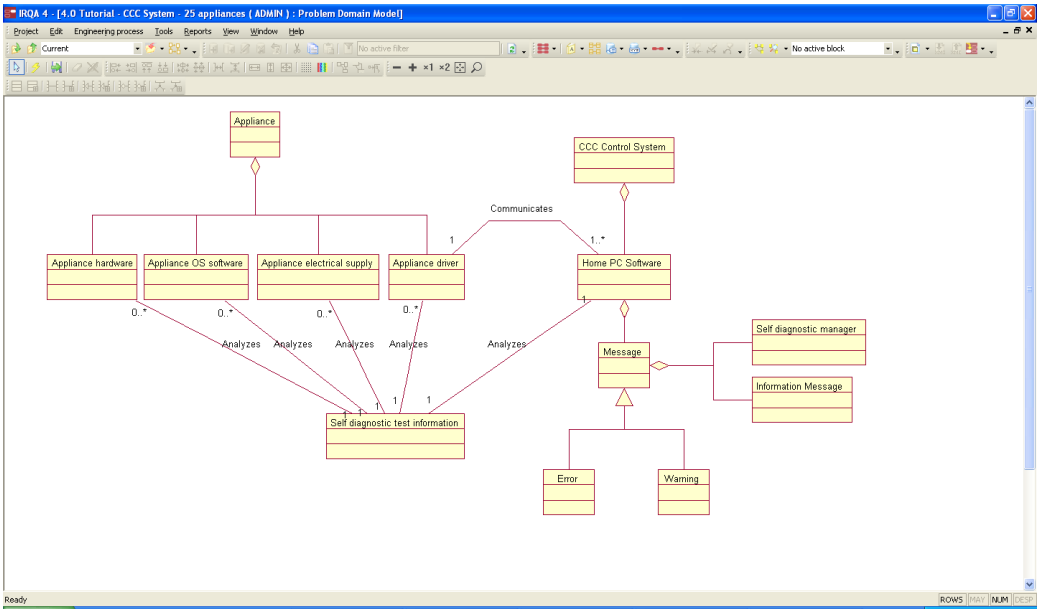


Problem Domain Models / Glossaries

As a Requirement Management & Definition tool, IRQA tries to cover more than just the mere management of requirements and go a step further. To help users with the elicitation of requirements making sure of the understanding of the domain IRQA includes the Problem Domain Model (PDM).

Through the PDM it is possible to define domain concepts to make sure that everyone understands the same. Those concepts may be characterized with attributes and also be related with other concepts. Finally, all this information may be represented in a concept diagram which is basically a UML class diagram.

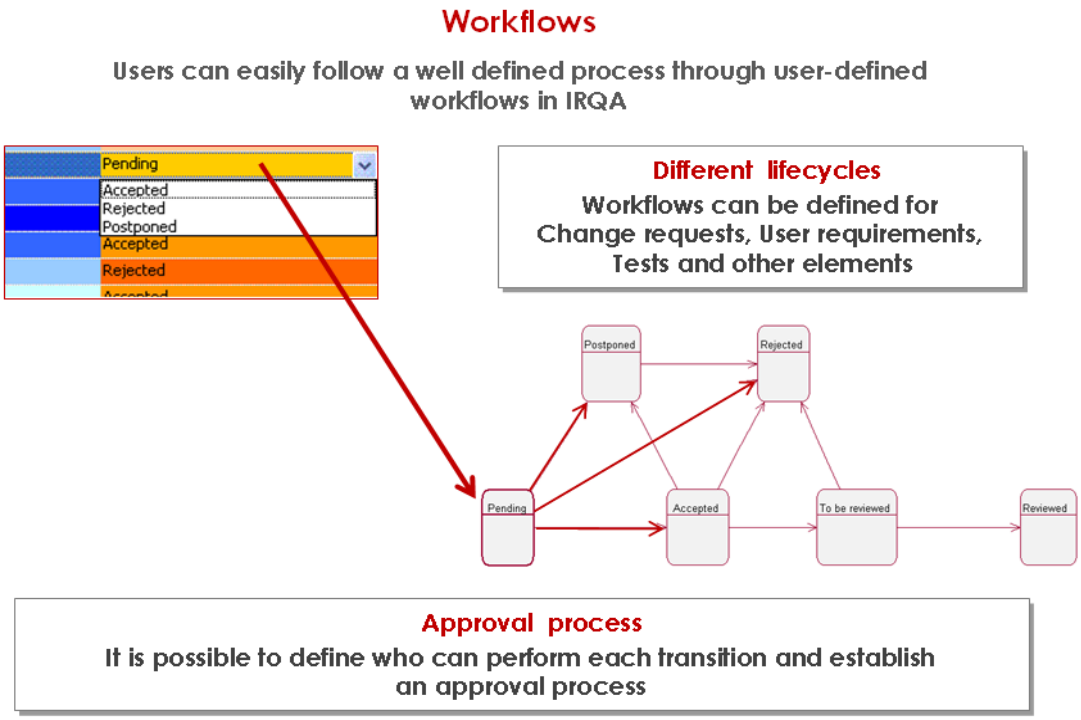
The requirements can be traced to those concepts that will be the origin of these requirements and can be also used as a checklist to write requirements.



Business Requirements & Solutions Lifecycle with Approval Workflow Embedded

As described before, workflows may be created into IRQA, different rights may be assigned to the transitions and finally, VB scripts may be added to the transitions so they will be executed every time a requirement switch the value of the workflow.

Baselines, on the other hand, are completely supported by IRQA. Not only total baselines (that will contain all the elements, relationships between them, attributes, etc.) but also partial baselines that will contain only those elements selected by the user (e.g. those who pass certain filter, like only the evaluated requirements). Baselines can be compared into IRQA also, with a system that will highlight the differences between each baseline. Finally, it is possible to mark a set of user that will need to sign electronically the baseline, approving it or rejecting it.

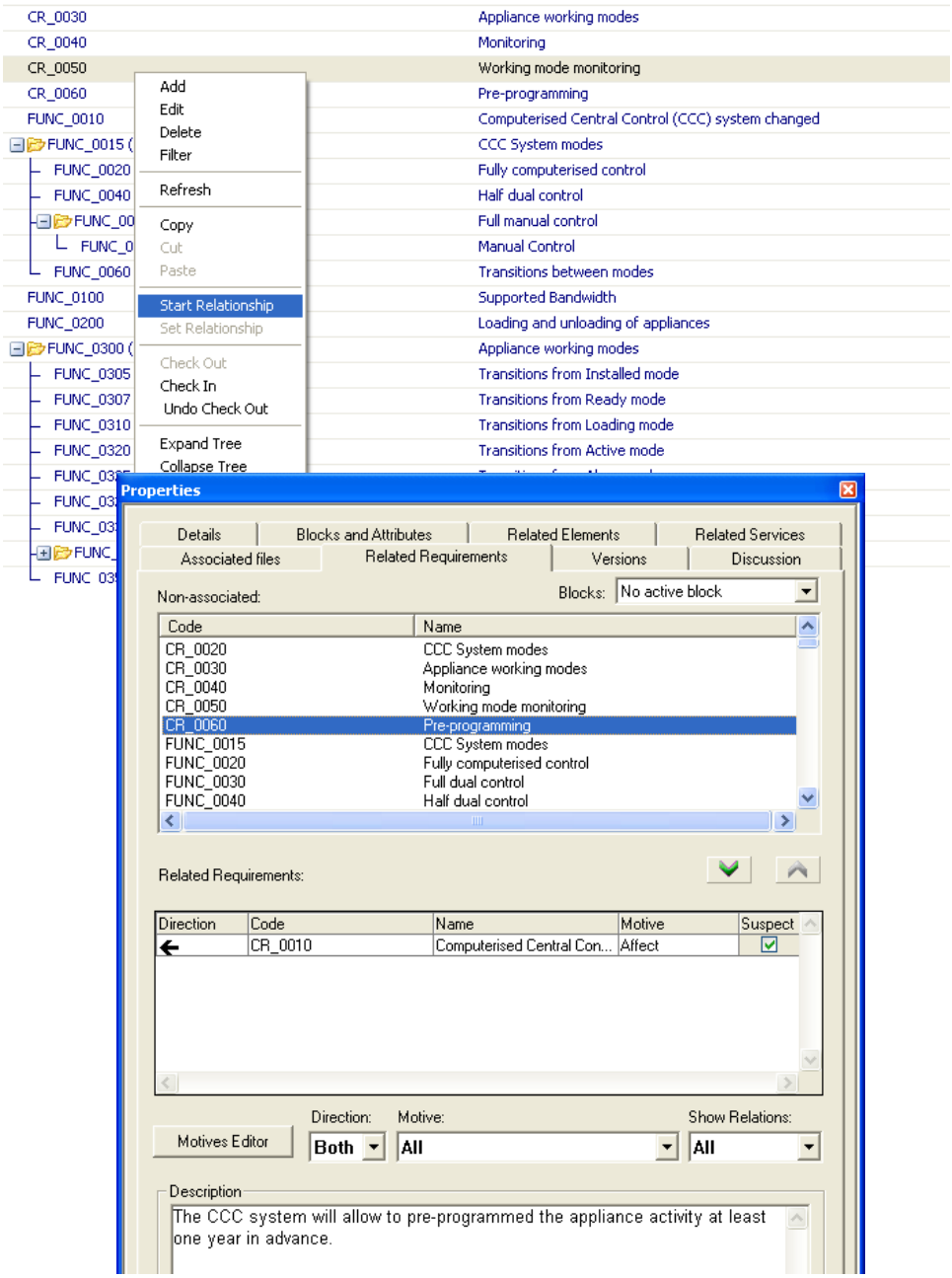


Business Requirements & Solutions Traceability

IRQA implements a complete traceability mechanism allowing the users to create traces horizontally and vertically. Those traces can be created/modified/consulted in both, a textual and a graphical way.

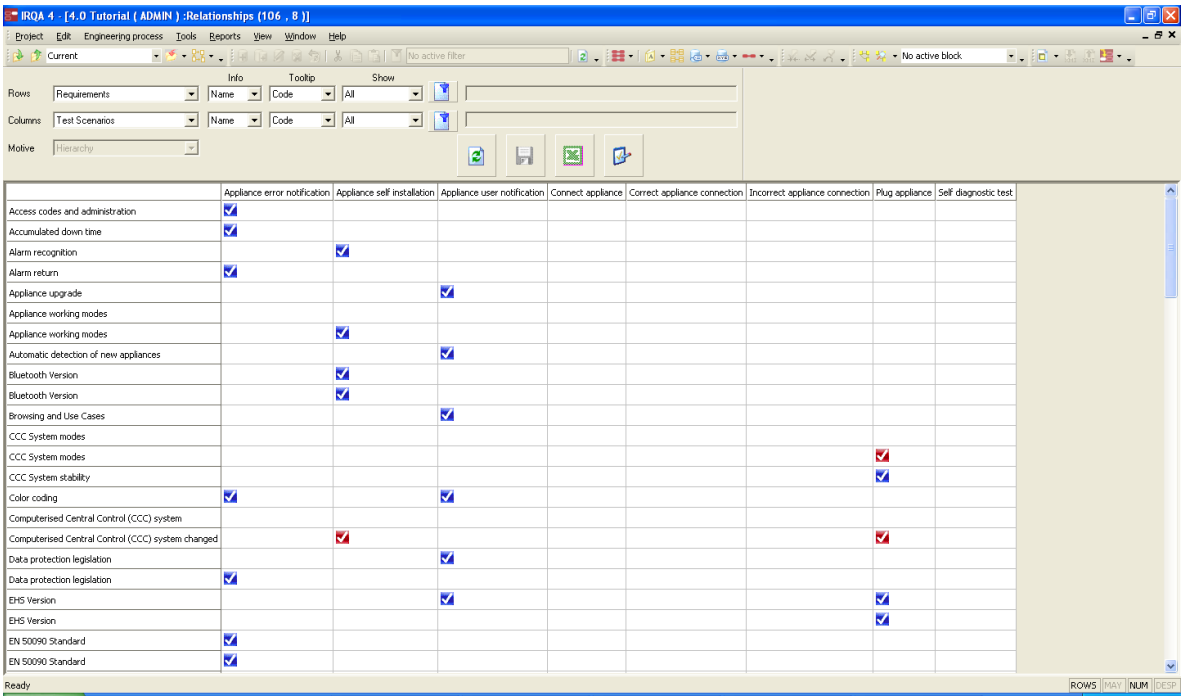
- Textual way:

  By selecting an element and tracing it with another one. Those traces can be established between requirements, requirements of different levels and requirements of different types or even between different element types within IRQA e.g. requirements with use cases, requirements with tests, and so on:



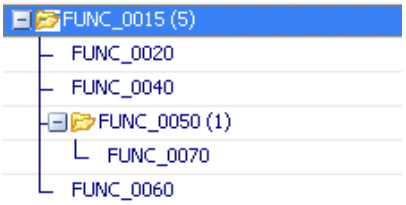
- Graphical way:

  Relationships can also be established through a traceability matrix. Those matrices show in rows and columns the information requested and allow the users to establish traces between those elements or modify the existing ones.
  Filter can also be applied over the elements in rows/columns making it easy to work with large matrix, for example, we can establish a filter to see only the system requirements that don't have any attached system requirements (those will be unfulfilled requirements)
  These matrices are a really powerful way to establish relationships but are also extremely useful to check the specification -as they can be used to ensure that the relationships are correct or that all the elements are related- or to perform impact analysis.

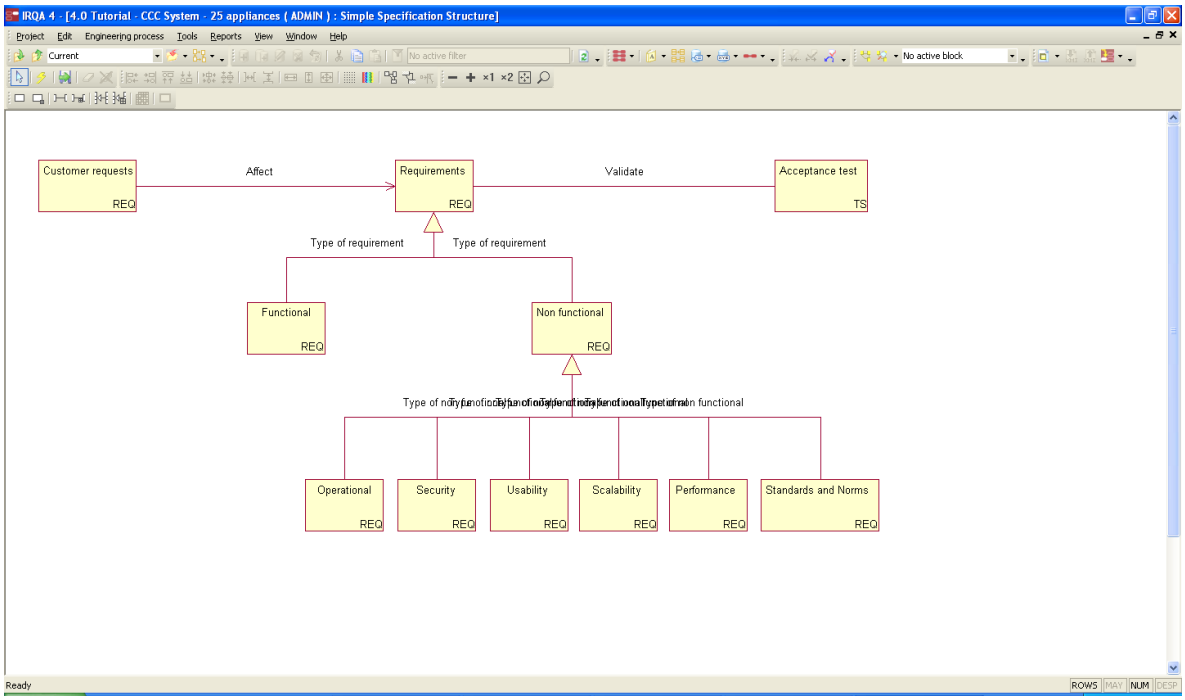


To make Impact analysis easier, IRQA implements the "Suspect Link" mechanism. If this option is turned on, every time we perform changes over an element (create a new version, modify the description, modify the attached files) all the relationships involving that element are turned into suspect as the tool is telling us that we should check the related elements and the relationships themselves as they may have some changes due to the modification. This can be used, for example, to see how the modification of a user requirement is going to impact in the system requirements that comes from that user requirement. Those suspect link can be checked graphically on the traceability matrix (they are shown as red marks) or textually (it is possible to filter to watch only the element with suspect relationships)

Additionally to these relationships, IRQA allow us to create hierarchical relationships, which represent relationships from more to less specific elements. Using those traces it is easy to create a hierarchy between requirements.



IRQA goes a step further that other RM Tools and let the users to restrict the available relationships. That can be achieved using a block diagram. Blocks are elements of the same type (requirements, tests, etc.) that share some common characteristics (for example: functional requirements, non-functional requirements, …) Using a block diagram users can represent the available relationships between elements within those blocks an even restrict that relationships so only the ones represented on the diagram can be established.
For example, in the diagram below, only Requirements may be traced with Acceptance tests.

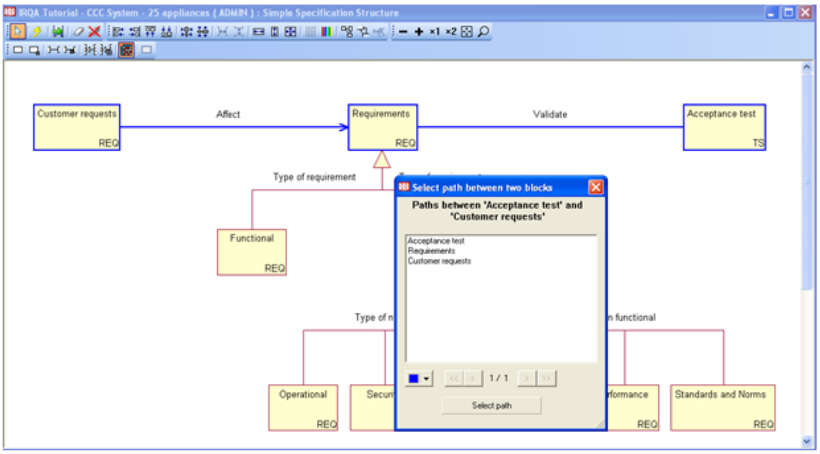


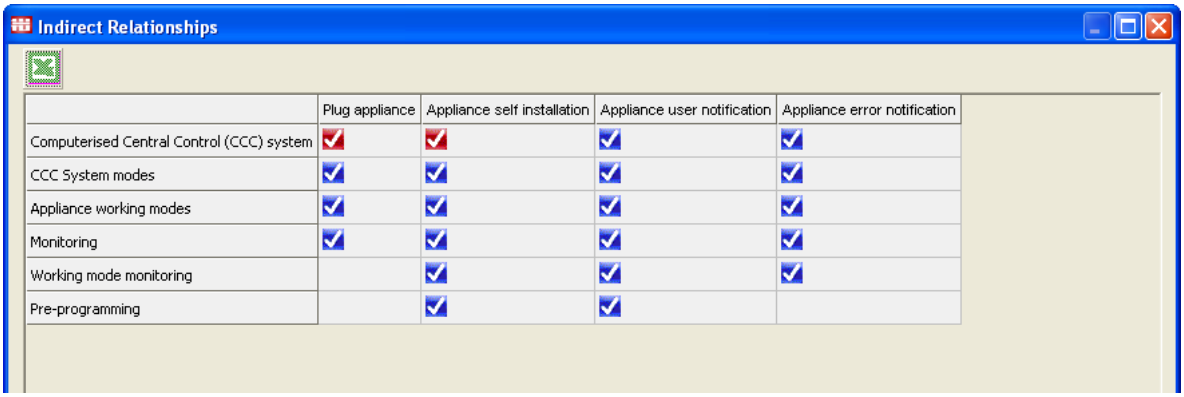
Additionally to those direct relationships, it is possible in IRQA, to check what it is called Indirect Relationships, e.g. in the diagram above, Customer Requests are traced with Requirements and those Requirements are traced with Acceptance Tests. It is not possible to establish relationships between Customer Requests and Acceptance Tests directly, but we can do that through the intermediate Requirements. It is possible to check that indirect traces (relationships between Customer Requests and Acceptance

tests through the intermediate Requirements) using an indirect Traceability Matrix. This matrix is telling us if all our customer requests are been tested through the intermediate requirements.

Indirect relationships

Block diagrams can display the impact between any pair of elements in our specification, even if they are only related through intermediate elements





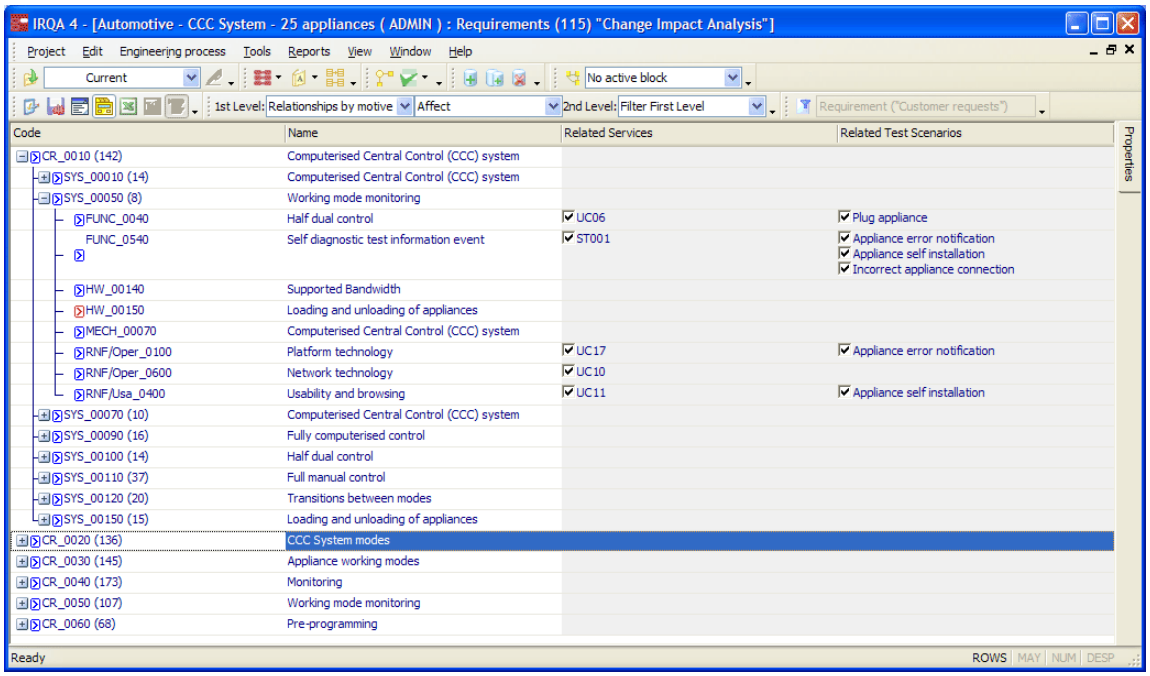
This information can also be achieved in a textual way using a Relationships View:

In this view users can see the horizontal traceability, that is, how high level requirements (usually user's requirements from the problem domain) evolve into more detailed requirements (in the solution domain). Additionally, the test related with those requirements and even the use cases attached can be shown in this view, allowing to the project managers to keep a whole vision of the project status.

Code	Name	Related Test Scenarios
CR_0010 (6)	Computerised Central Control (CCC) system	
CR_0020 (11)	CCC System modes	
FUNC_0305	Transitions from Installed mode	
FUNC_0340	Event generation	Appliance error notification
FUNC_0440	Pre-programmed states	
FUNC_0540	Self diagnostic test information event	
FUNC_0645	Protocol and data conversion	
FUNC_0655	External networks support	Appliance user notification Plug appliance
FUNC_0665	Local networks support	
NF/Std_0115	Java Version	Self diagnostic test
NF/Std_0200	Home security legislation	Correct appliance connection
RNF/Usa_0000	System interface	
RNF/Usa_0700	Browsing and Use Cases	Appliance user notification
CR_0030 (10)	Appliance working modes	
CR_0040 (7)	Monitoring	

Requirements Flow-Down

It is possible to create several requirement levels (from user requirements to more detailed system requirements) each of them with their special characteristics and attributes, as described graphically in the block diagram. The traceability between all these artifacts is guaranteed in IRQA. Additionally, IRQA allows the traces between requirements and external files that can be even into a SCM software which will allow us to trace requirement even with design element or even implementation class or functions.

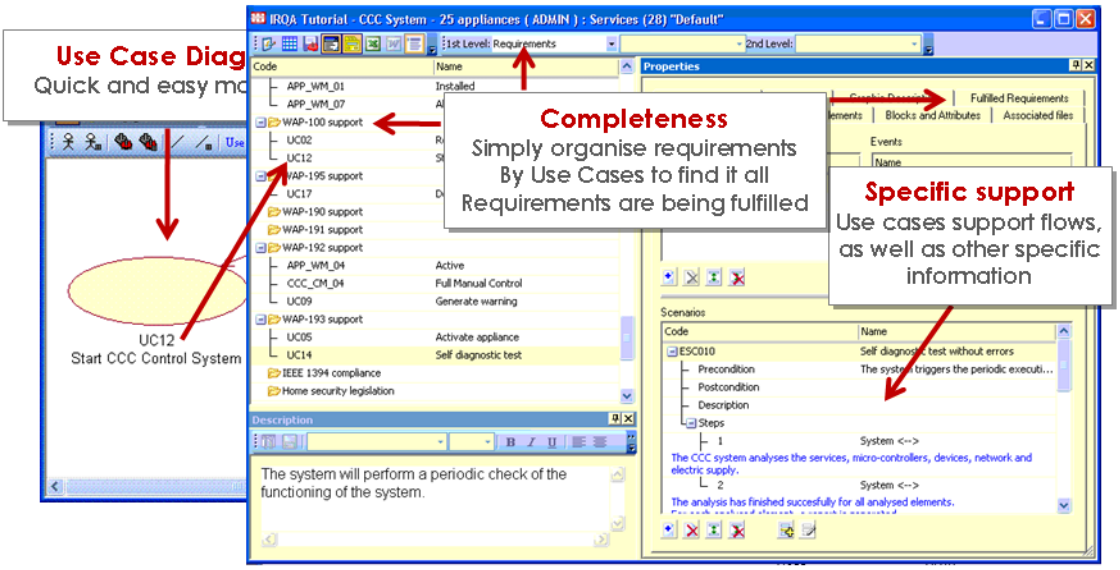


Being IRQA a requirements tool we keep those requirements communicated with other artifacts of the lifecycle. With this in mid, IRQA supports the definition of Use Cases and

also Tests Cases (for validation/verification purposes) that can be defined into IRQA and also traced between them.

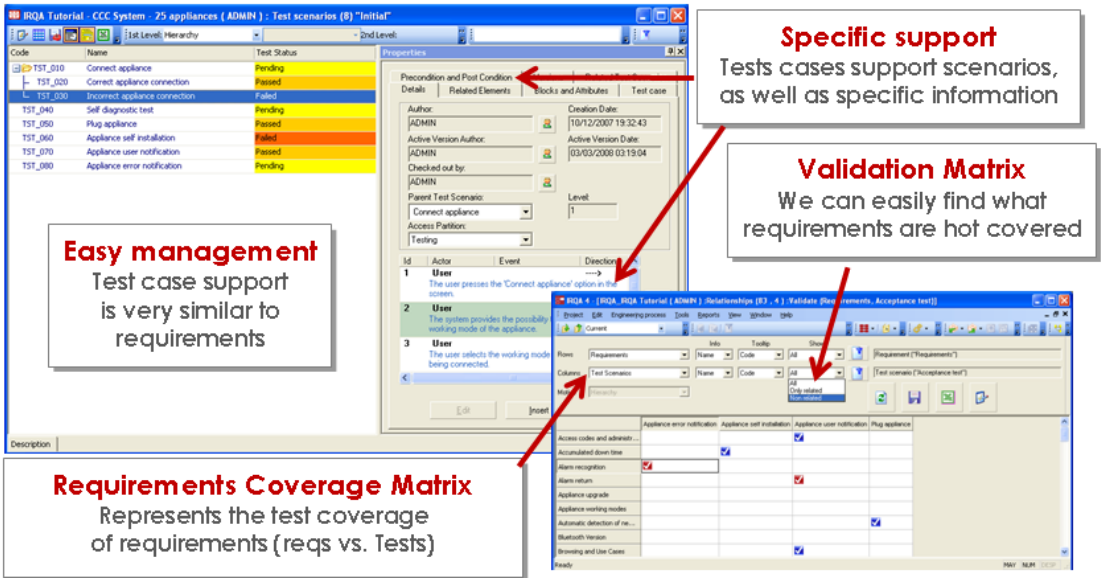
Use Case support

Use Cases provide a simple and powerful way to communicate with, and gather requirements from, stakeholders. It also helps clarify requirements and create a common understanding between all development team members and stakeholders, fostering communication and collaboration across all stakeholders



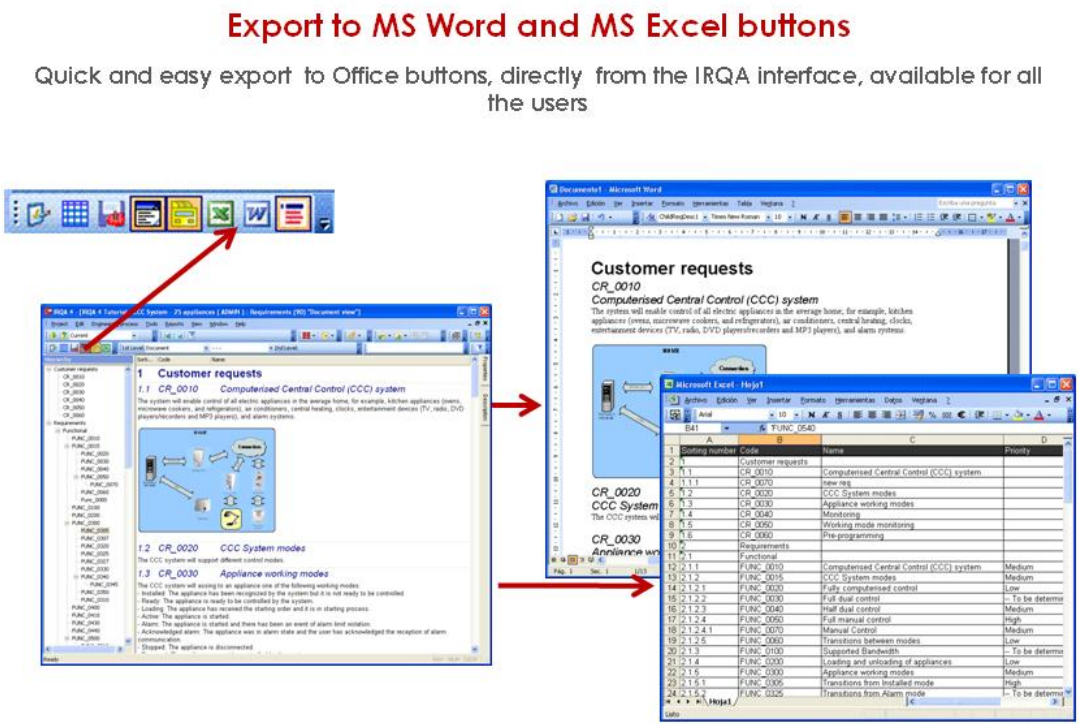
Acceptance Tests support

Requirements should be validated through test cases, which are specifically supported in IRQA. One requirement can be traced to one or more tests, as well as a test can be traced to one or more requirements

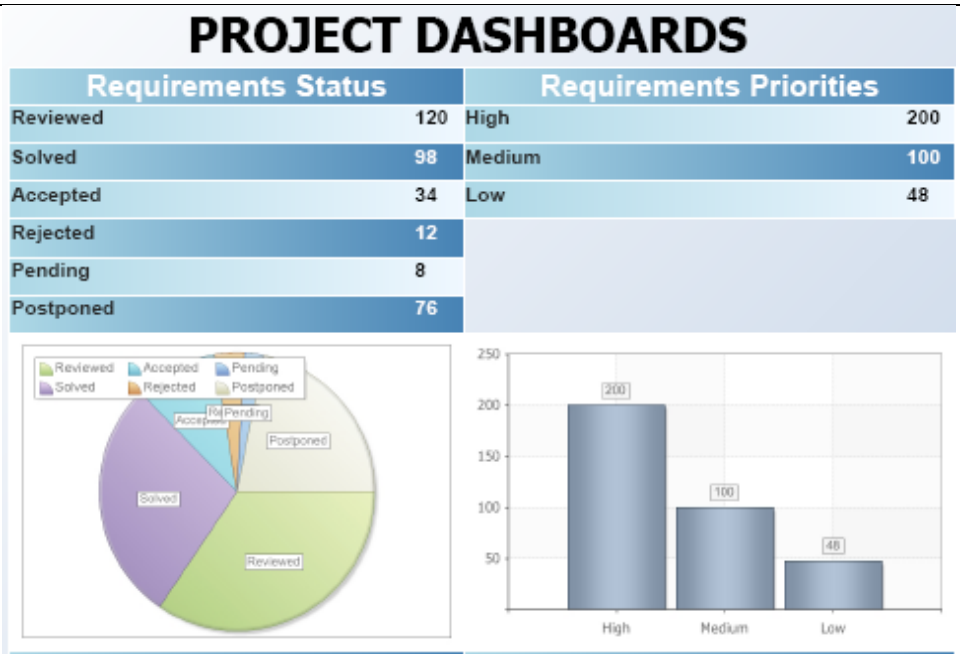


Documents and other Output Media

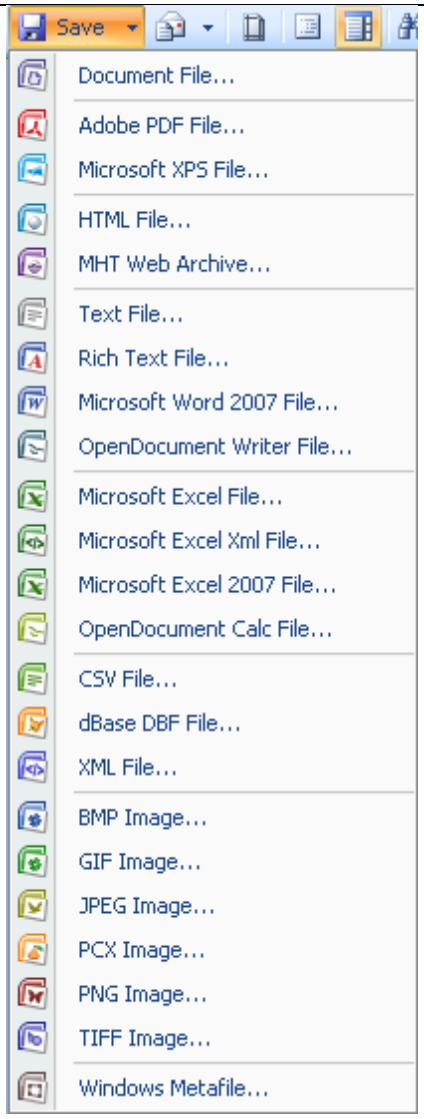
IRQA support what-you-see-is-what-you-get type of export. These exports will export the elements currently displayed in the view with their corresponding attributes.



However, for more formal document it is possible to generate reports from IRQA. The tool includes a set of these predefined reports with a more structured format (headers, footers, front pages, etc.) that will satisfy the basic reporting needs most users. Additionally, for more detailed reporting needs, IRQA allows also the creation of new reports. These reports may contain the needed info from IRQA with your own fonts, images, headers, graphs, chars, etc. Even programming may be added to these reports so it is easy to get statistics and metrics from the report.



There is a set of predefined reports in IRQA. Those reports try to cover basic reporting needs that the users may have. Once generated, these reports may be exported to several external formats, including PDF, MSExcel, MSWord, JPG, BMP, HTML, CSV and XML, between other.



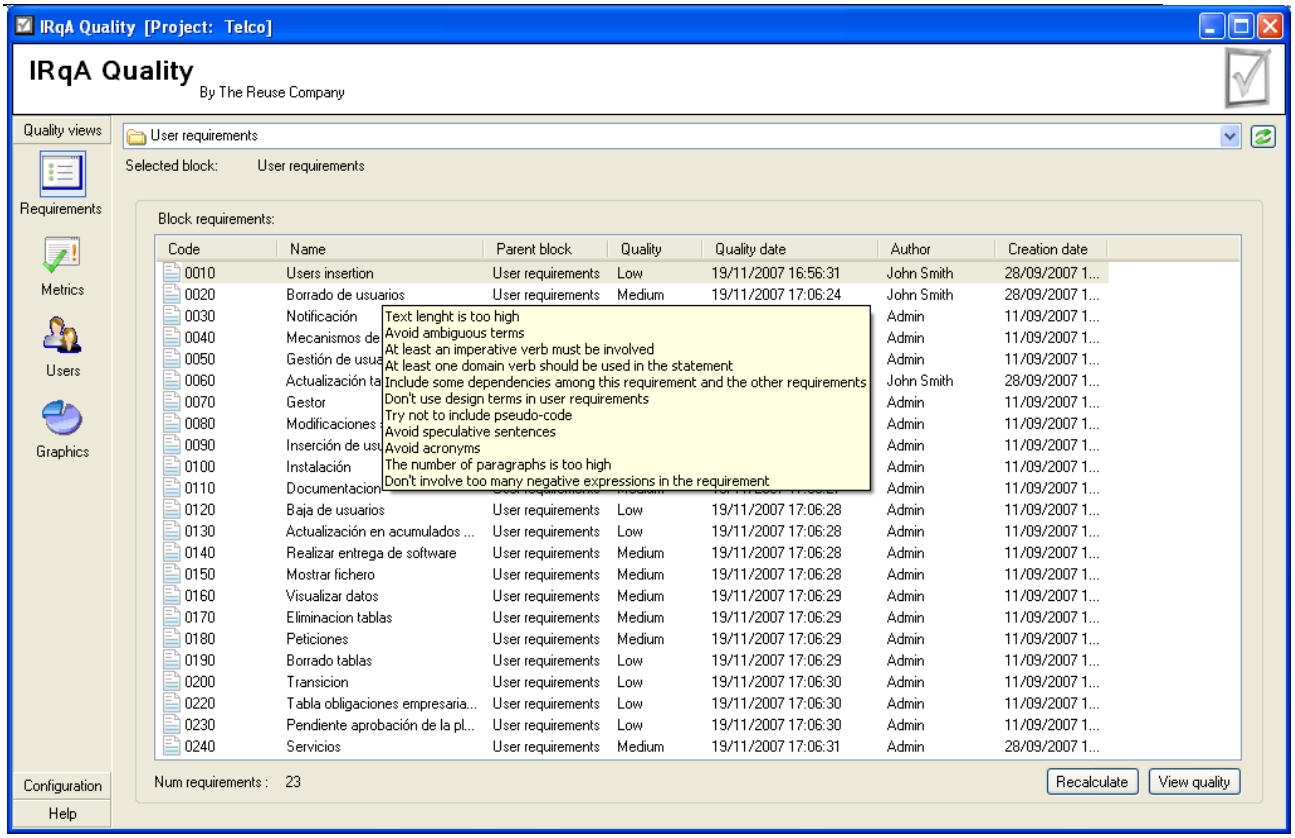
Additionally to these predefined report, the IRQA Report Manager may be used to modify the existing reports or to create new ones. Using this tool it is possible to create a report containing the information you want (almost any information within IRQA can be used to generate the report) and with the format (fonts, images, headers, styles, etc.) you want. These reports can also contain metrics obtained from the data within IRQA, for example you can generate a report that tells the status of the requirements and at the end of the report generate a bar chart with the different number of requirements in the different statuses.

Additionally to these IRQA reports, if some quality metrics are needed, they can be obtained using the IRQA Quality Analyzer, a semantic analyzer that is going to measure the quality of the requirements in the specification based on certain pre defined rules that are fully customizable by the administrator. This tool will generate graphs and statistics with the quality of our requirements and additionally will generate some

reports with advices to improve the quality of our reports. There are several rules including:

Metric	Description
Ambiguity	Ambiguous terms make requirements difficult to understand and, thereby, difficult to test
Conditional	Conditionality must not be explicit in the text of the requirement. If needed use an attribute
Design Sentences	Design terms could lead to over-specification
Domain Terms	If you involve too many domain terms in the requirement, maybe it's expressing more than one need
Domain Verbs	At least one domain verb must be involved, but may domain verbs imply many needs within the same requirement
Flow Sentences	Flow terms could lead to over-specification
Imperatives	Requirements should be conjugated using the imperative: - Requirements use shall - Statements of fact use will - Goals use should Nevertheless, you should not include many verbs in every requirement
Incompleteness	Incomplete terms makes the requirement non-atomic
Negative Sentences	More than two negative terms in the same requirement makes it difficult to understand
Speculative Sentences	A speculative term make the requirement difficult to test
Acronyms	An acronym, if it's not included into the domain, may not be understood by some stakeholders
Chars between punctuation	If you write long sentences without punctuation marks, the requirement will be difficult to understand
Connectors	Too many connectors in a requirement may imply different needs into the same requirement
Dependencies	A requirement with so many dependences could be difficult to understand
Implicity	Too many pronouns make the requirement difficult to understand
Nested levels	Too many nesting levels makes the whole specification difficult to understand
Paragraphs	A requirement should not be written using too many paragraphs
Readability	Long words and long sentences make the requirement difficult to understand
Subjectivity	You shouldn't express your will into requirements
Text Length	The length of a requirement should be enough just to state what is expected out of the user need

These rules are totally customizable to contemplate company specific needs and terms, as well as configure them for each type of requirement in the project.



User Interfaces

As previously said, IRQA interfaces are fully customizable to each user, so it is possible to create a specific user interface for each user group.

IRQA supports multitask so it is really simple to, for example, generate a report while you are working with the requirements or import a set of requirements while you are modifying the specs.
The tool GUI can display the information in the same or in different windows, in order to manage complex information in the simplest possible way. For example, two different projects may be opened at the same time and different views of the same project may be managed in parallel.

A change performed in IRQA will automatically update the elements so if other users are working with them or other views are using them they will get the latest changes by just refreshing (auto refresh is also available)

As said before, IRQA is not just a textual tool so it is possible to input data through diagrams (use case diagrams, DFDs, State diagrams, sequence diagrams, block diagrams, class diagrams, etc.).

Data input in IRQA through the user interface includes all controls needed to prevent erroneous data from getting into the data base.

MS Windows copy and paste functionality is allowed for all text fields in the tool, and also in order to duplicate elements in an IRQA project or even to copy elements between different IRQA projects.
IRQA is a Microsoft Windows application following the look & feel of standard Microsoft applications.

Data input in IRQA is written in the data base directly after being entered .

Finally, Undo is supported in requirements through the undo check out option. IRQA uses a check-in/put mechanism similar to the one used by the SCM tools. A user needs to check-out a requirement in order to work with it, this will block the requirement exclusively for him (other users may still read it, but not change it). Once all the changes have been performed, the user must check-in the requirement making it available for other user to check it out. This check-in/out cycle will additionally create a new version of the requirement. A user can decide at any moment that he wants to undo a check out which will undo all the changes in the requirement and restore it to the previous version. Apart from this, undo while working with the description is supported just by clicking Ctrl + Z (just like MSWord).

Standards

Several standards are used into IRQA:

Database standards:

- ✓ For SQL Server repositories, SQL Server
- ✓ ORACLE
- ✓ It is not necessary to install any MS ACCESS version in order to use ACCESS repositories.

API Standards

IRQA comes with an open API that support the following languages:

- ✓ IRQAFace: COM API, for COM objects: C, C++, C#, .Net, VB, etc.
- ✓ J-IRQAFace: For Java

Plug-ins

IRQA comes with a complete plug-in SDK that allow you to create your own plugin in C/C++

Modeling

UML notation, Functional notation (E/R diagrams, DFDs)

Output documents

Reports may be produced in a variety of industry standard formats or user-defined formats.

Exchange data standards:

Several external formats are supported for exchanging data or simply to generate documentation from IRQA:

- XRI (XML for requirements interchange)
- XMI
- XML
- MSWord
- MSExcel, CSV
- PDF
- HTML
- Images (jpg, bmp,. Etc.)
- RTF
- OLE Objects (supported in the description field)

SCC Protocol:

Using this standard it is possible to link requirements with files stored in a SCM tool (like Subversion o SourceSafe)

