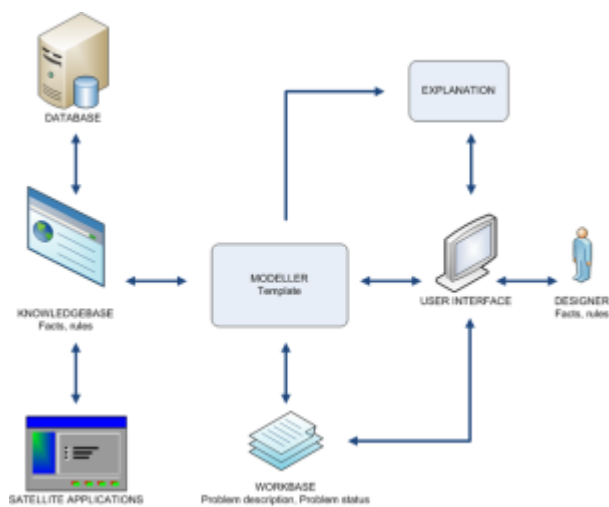


What is Quaestor

- [Advantages](#)
- [Technical details](#)
 - [Knowledge base development](#)
 - [Model development](#)
 - [Results viewer and simple model use](#)
- [License options](#)
 - [A. User modes](#)
 - [B. User levels](#)
- [Papers about Quaestor](#)

Quaestor is a software framework for developing and using knowledge based systems containing computational methods and data. With these systems, you are able to streamline analysis, design and engineering processes.

The general architecture can be sketched as follows:



The combination of a project (a project file), a knowledge base (a knowledge file) and the Quaestor program makes a knowledge-based system to streamline design and engineering processes (workflows).

Advantages

Knowledge-based systems in Quaestor enable research, engineering and design departments to simplify the tasks of:

- managing computational methods and model fragments;
- manage and use of complex (and knowledge based) workflows in addition to data and calculation management;
- controlling analysis, design and engineering processes;
- performing goal driven design, engineering and mathematical model development;
- project administration (all information from project data to calculation results and reports in one environment).

Next to the above features, Quaestor provides a flexible computational environment that allows:

- the concurrent use of various computational models (formulas; rules; programs; spreadsheets);
- computing solutions for problems with one or more varying independent parameters;
- easy maintenance and rapid development of new applications by including new model fragments in a knowledge base;
- the use of legacy software, prediction tools, databases, spreadsheets and relations.

This makes Quaestor one of the most powerful environments currently available for the development and use of knowledge-based computational models, applications and configurators.

Technical details

Quaestor can be used in three ways, depending on your personal requirements and user license. See [user levels documentation](#) for more detail:

1. As an application development and knowledge management platform (knowledge engineering);
2. As an user platform (domain expert use)
3. As a results viewer (end use),

Knowledge base development

A Knowledge Engineer (KE) can store computational model fragments in a Knowledge base. These fragments consist of relationships, formulas or tools with their validity, static data, descriptions, optional illustrations, properties and parameters. Since the output of one model fragment can be input for another, all components together form an undirected network of 'rules'. This network controls the data and calculation management capabilities of the system.

In addition, a Knowledge Engineer can create a Taxonomy of a process or product. This is the blue print (or DNA) of the process or product. In this way, the KE is able to use the workflow capabilities of the system. It enables the KE to have full control over the process or workflow during the analysis, design or engineering process without losing the power of the Quaestor model assembly capabilities. All parameters, relations, connections to other programs etc. can be fully described in text, illustrations and hyperlinks to further background information. In this way by developing knowledge bases, you are managing all your knowledge on methods and processes used in your environment.

Quaestor applies a unique and advanced bi-directional chaining strategy of reasoning, computer algebra as well as network and relational database concepts. It offers a knowledge management environment and is able to invoke external executable programs, spreadsheets and databases.

Model development

A Domain Expert (DE) and Knowledge Engineer can use a knowledge base (network of model fragments) to determine or optimise any parameter used in any of the available formulas or model components (knowledge). This process is called Model Assembling and its result is a Solution.

Model assembly starts with the selection of one or more parameters as top goal(s). The user provides the input data and makes the choices required to solve a problem in a dialog session. Formulas or methods are suggested and the user is prompted for input values of parameters applied in the proposed relations. Quaestor will attempt to compute any value that is required and not provided as input by searching for suitable model fragments available in the knowledge base. By either accepting or rejecting suggestions, the user selects rules and thus fully controls the way the top goal(s) are achieved and a model is assembled.

Note that the knowledge and relations in the knowledge base determine the process order. Although a very powerful asset of Quaestor, it might sometimes lead into unexpected information requests. For this reason the taxonomy technology is introduced. In addition to the above classical use, users can start a configuration process based on this structure. The structure uses the same characteristics as described above. The difference is that you can either follow the order as defined in the taxonomy or go through the tree anyway you want. Quaestor will manage and communicate about parts of the process that require information or should be calculated to finalize your process or product. The result is called a Taxonomy Solution.

Powerful document generation and browser facilities are available to present results, get information on intermediate calculations, the reasoning process and knowledge used in the solution.

Results viewer and simple model use

Any modeling process results in a Solution. This Solution can be viewed in every detail defined by the Knowledge Engineer and can also be restarted using your new input.

Users can always select the "Standard" solutions created during the model development that will act very much like normal applications. End Users can only restart Taxonomy Solutions as long as nothing will be changed to the structure of the solution.

Please note that in addition to providing single values for parameters, the user can provide ranges of values for several parameters. This makes it possible to compute matrices of parameter variations to perform trade-off studies or to check the robustness of solutions.

License options

The software can be purchased with an annual, non-exclusive, non-transferable, user license. A demo version of Quaestor is available for free. This version provides full user rights up to 100 frames per knowledge base (i.e. 100 parameters, relations and constraints in total).

The fee for an annual user licence will depend on (A) the user modus of the software (stand-alone or concurrent use) and (B) the user level (End-User, Domain Expert, Knowledge Engineer).

A. User modes

You can use Quaestor systems in two modes:

1. Stand-alone:
Quaestor is installed on every individual PC with its own licence and activation string.
2. Concurrent use:
Quaestor is installed as client version on any PC in a Local Area Network. Additional software (the Qknowledge Licence Manager) manages the concurrent licenses. Concurrent user licenses are more expensive but require a smaller number of licenses, depending on the desired availability of the system.

B. User levels

Quaestor systems can be used in two levels:

1. [End-user](#) or run time licence:

In this case Quaestor is usually part of a specific application (knowledge base) and can only be used with this application. Knowledge and models can neither be changed or added, nor removed. Without the application (knowledge base) Quaestor can be used as demo version.

2. [Domain expert](#):

In addition to an end-user, a "Domain expert" (DE) is allowed to assemble any model (Solution) using any (protected) knowledge base. Knowledge bases protected by knowledge engineers can sometimes still only be used as End user.

In addition you have a development level:

[Knowledge engineer](#): A "Knowledge engineer" (KE) has full rights to develop and maintain knowledge bases. Knowledge bases protected by other knowledge engineers can still only be used as Domain expert or End user.

For more details see also: [User levels and rights](#)

Papers about Quaestor

These articles give more insight and show the power of Quaestor.

- [Knowledge Based Computational Model Assembling]
- [HME Innovation Award Qnowledge](Dutch)
- [Schip & Werf](Dutch)