

software development

February 2004

DESIGN ▶ BUILD ▶ TEST ▶ DEPLOY

Bringing Order to Chaos



Stow the whiteboards: Telelogic's DOORS 7.0 lets you manage requirements with the same level of sophistication with which you manage your source code. **BY ALLAN MCNAUGHTON**

MANY A PROJECT MANAGER HAS SKATED by without using a requirements management tool—often from fear of bogging down the development process with excessive formality. Yes, you can certainly use Word, Excel or even the back of a napkin for tracking requirements, but on anything other than the simplest project, you'll soon be flummoxed, wondering what version of a change goes where and just how it affects everything else. Tools like Telelogic's DOORS 7.0 Enterprise Requirements System can bring order to this chaos. You wouldn't consider running a project without a source-code configuration system, so why not apply the same approach to requirements management?

Project managers should look for a requirements management tool that meets the following criteria:

- ▶ Ability to import requirements in multiple formats (so you can start working immediately regardless of how you were originally tracking requirements).
- ▶ Support for traceability and impact analysis (so you can quickly tell how a change in one requirement affects other parts of the system).
- ▶ Flexible access controls and a malleable change management system (so you can give power to those who need it, restrict access to those who don't, and institute an effective change management process).
- ▶ Support for baselines and electronic signatures (so you can capture a project's historical state and prove who signed off on it).
- ▶ Flagging of modifications to the requirements database (to avoid wading through a thousand requirements looking for

Telelogic DOORS 7.0 ERS

Rating: ★★★★★

PROS

1. DOORS is powerful, flexible and easy to use.
2. It has excellent linking and reporting capabilities.
3. It has strong versioning and change-management tools.

CONS

1. It lacks bidirectional synchronization with Microsoft Word.
2. The Web interface is sold separately.
3. The price tag may be too high for smaller shops.

CONTACT INFORMATION

Telelogic
9401 Jeronimo Rd.
Irvine, CA 92618
Tel: (877) 275-4777
Fax: (949) 830-8023
www.telelogic.com

TECHNICAL REQUIREMENTS

Microsoft Windows 98 (SEO, Windows 2000 Professional, Windows 2000 Server, Windows NT 4.0 Service Pack 6a or Windows XP, Hewlett-Packard HP-UC B.11 (32-bit and 64-bit versions) and Sun Solaris 8 or 9.

what changed last).

DOORS 7.0 ERS, a fat-client application that runs on Microsoft Windows and Unix systems, forms the foundation of Telelogic's requirements management product line, which is complemented by DOORSnet (a browser-only version that extends DOORS to those who need only to occasionally review or update requirements) and DOORSrequireIT (a Microsoft Word-based tool expressly designed for nontechnical end users).

Getting Started

With its straightforward client/server architecture, DOORS is a

Allan McNaughton, a long-time writer and developer, is the principal analyst at Technical Insight LLC. He can be reached at allan@technical-insight.com.

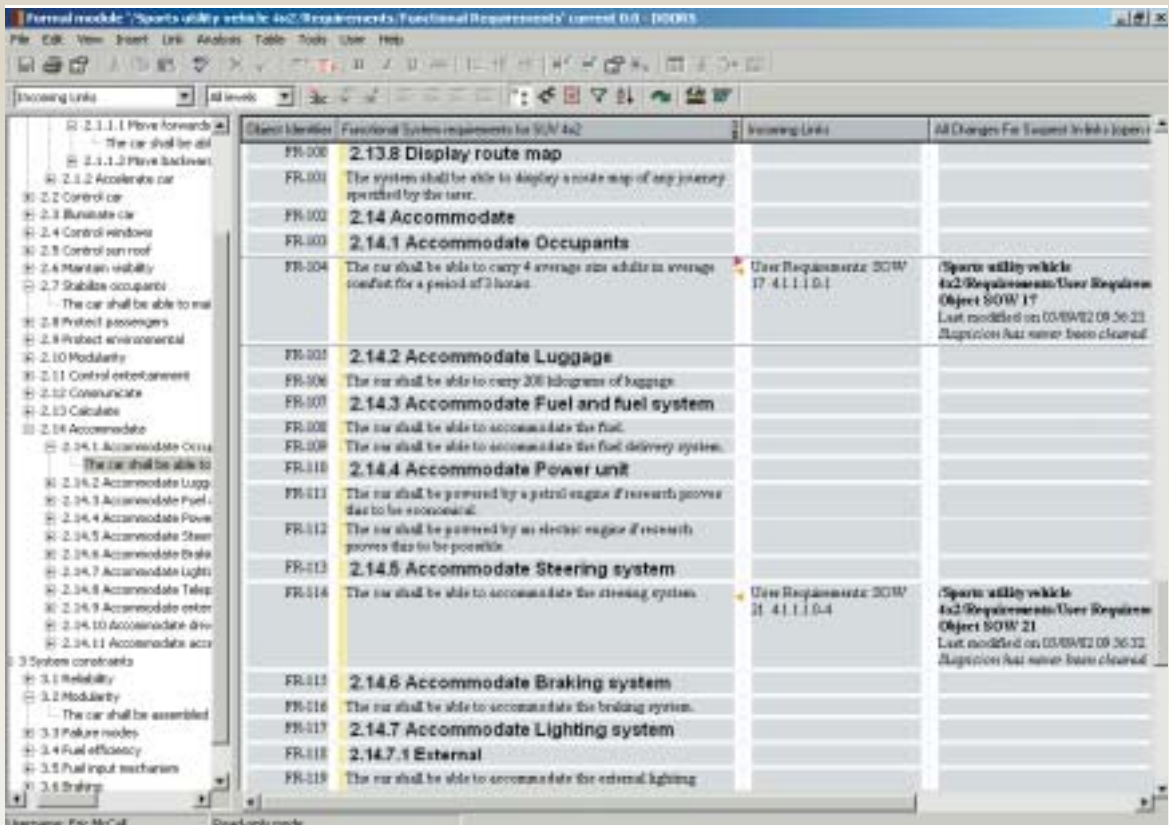
breeze to get up and running—even on Unix. Simply follow the well-documented installation steps on your server, then on your client machine, configure your license, create the list of users (or import them from an LDAP server), define the project structure using their wizard, and voilà—you’re ready to input requirements.

Populating DOORS is a matter of entering requirements using its word processor-like interface, or importing them from another source. If your requirements are already in a Word, Excel, Microsoft Project, FrameMaker, Interleaf, RTF or plain text file, DOORS provides a wizard that speeds the process. Although DOORS can import (and export) requirements in multiple formats, it doesn’t support bidirectional synchronization of subsequent changes with the original data source; that is, once you manage your requirements in DOORS, you’ll always manage them in DOORS, unless, of course, you export them to a neutral format and then import them into another tool. (DOORSrequireIT, sold separately, solves part of this problem by synchronizing with Microsoft Word.) Although not available for this review, a planned addition to DOORS will let you create and synchronize Microsoft Project 2003 data from an existing set of requirements.

DOORS doesn’t treat requirements as structured text, but as distinct objects with configurable sets of attributes. Once your requirements are in DOORS, they can be manipulated in many ways—much as with a relational database. For example: Each requirement object exists within what DOORS calls a “module” (similar to rows in a table, although they reside in a proprietary database so that you can’t access them outside of DOORS). You might create separate modules to manage user requirements, software design requirements or test plan requirements.

The Power of Links

DOORS can function as a stand-alone requirements tool, but that’s like talking to a wall: Linking requirements to the development process enforces accountability and gets everyone’s ear. DOORS can create links with other popular tools, such as Rational’s ClearCase, ClearQuest or Rose, Merant’s PVCS and Mercury Interactive’s TestDirector. These integrations tighten the link between a product’s requirements and the software under development. For example, in the case of TestDirector, you can link requirements in DOORS to a TestDirector project and transfer test details, results and defects from TestDirector back to DOORS.



Functional Requirements for SUV 4x2

Object Identifier	Functional System requirements for SUV 4x2	Viewing Links	All Changes For Selected Inherited Objects
FR-000	2.13.8 Display route map		
FR-001	The system shall be able to display a route map of any journey specified by the user.		
FR-002	2.14 Accommodate		
FR-003	2.14.1 Accommodate Occupants		
FR-004	The car shall be able to carry 4 average size adults in average comfort for a period of 3 hours.	User Requirements: SUV 17 4.1.1.0.1	Sports utility vehicle 4x2/Requirements>User Requirement Object SUV 17 Last modified on 03/04/02 09:36:23 Suppression has never been cleared
FR-005	2.14.2 Accommodate Luggage		
FR-006	The car shall be able to carry 200 kilograms of baggage.		
FR-007	2.14.3 Accommodate Fuel and fuel system		
FR-008	The car shall be able to accommodate the fuel.		
FR-009	The car shall be able to accommodate the fuel delivery system.		
FR-010	2.14.4 Accommodate Power unit		
FR-011	The car shall be powered by a petrol engine if research proves that to be economical.		
FR-012	The car shall be powered by an electric engine if research proves that to be possible.		
FR-013	2.14.5 Accommodate Steering system		
FR-014	The car shall be able to accommodate the steering system.	User Requirements: SUV 21 4.1.1.0.4	Sports utility vehicle 4x2/Requirements>User Requirement Object SUV 21 Last modified on 03/04/02 09:36:23 Suppression has never been cleared
FR-015	2.14.6 Accommodate Braking system		
FR-016	The car shall be able to accommodate the braking system.		
FR-017	2.14.7 Accommodate Lighting system		
FR-018	2.14.7.1 External		
FR-019	The car shall be able to accommodate the external lighting.		

Moving Up, Moving Down
DOORS annotates each linked requirement with a small glyph that shows the direction of the link (incoming or outgoing).

Links give you traceability, letting you demonstrate that what you're building satisfies user requirements. For example, the project manager can follow links between requirements modules to see that the engineering team fully supports a user requirement and that the QA team plans to test what's actually being built. You can follow links in both directions of the requirements hierarchy. If a test fails, you can find out which requirements are affected by tracing the links from the test back to the design features, and from the design features back to the user requirements. Links allow you to effectively manage change. With an appropriate set of links, you can quickly trace the impact of a change to a single requirement on the rest of your system.

Given the importance of links, DOORS provides many tools to create, manage and report on them. Creating links is a snap: Simply drag and drop one requirement onto another, or mark the first link as the source and then denote the second as the target. DOORS annotates each linked requirement with a small glyph that shows the direction of the link (incoming or outgoing); the glyphs can be used for navigating from one linked requirement to another (See "Moving Up, Moving Down," page 29). Moving your mouse over the glyph also reveals the name of the requirement at the link's destination. However, this feature works well *only* when DOORS has already loaded the requirements module that contains the destination (that is, you've already viewed a requirement there during the current session); otherwise, all you see is the name of the destination module and a not-very-helpful numbered link (such as UserRequirements/1.3.5.1).

With DXL, the DOORS eXtension Language, the sky's the limit with regard to how DOORS can be customized.

To experiment with DOORS' link analysis capabilities, I created a set of incoming and outgoing links between the different requirements modules in my DOORS project. I constructed links that connected individual user requirements to their corresponding use cases, and created other links to relate a group of verification requirements back to their appropriate system requirement (a one-to-many relationship). With these links in place, I could then use DOORS' impact and traceability analysis tools.

The impact analysis tool let me see which objects in the current module's outgoing links would be affected if changes were made to that module. Conversely, the traceability analysis tool displays information about the objects in the current module's incoming links. The creation of these useful reports isn't auto-



Finding Your Way

DOORS includes a traceability explorer, which lets you see at a glance which objects in the current module have links and allows you to navigate to the linked objects.

matic, however: The user must explicitly request their generation. Keeping this information up-to-date and available by default would be a real plus.

Another word of warning: Finding all the links can be painful when working with a large requirements document. To streamline this process, DOORS includes a traceability explorer, which lets you see at a glance which objects in the current module have links and allows you to navigate to the linked objects (see "Finding Your Way," this page). Although the traceability explorer does ease the task of seeing what links exist within a requirements document, it doesn't provide a big-picture overview of the relationships (links) among a set of requirements documents.

A Clear View

On a large project, it's easy to be overwhelmed by the sheer quantity of requirements and their corresponding links. The concept of creating *views* of interest (through filtering and sorting) is central to DOORS: It provides a highly configurable report window into the requirements database. With views, you can cut the list of requirements down to size, select which attributes to display, choose how to display them, and determine who gets to see them. For example, you can create a view that shows specific attributes of only those requirements that contain a matching word (or regular expression) and are linked to requirements in another module. Once you've created a view, it can be given a name and saved in a list for later use by other members of your team.

The process of customizing a view also gives you a glimpse into the power of the DOORS eXtension Language (DXL). DXL

is a C-like scripting language (with auto-declare variables) complete with support for features such as derived types, file streams and OLE objects. You can also use it for simple things, such as computing the difference between two date attributes for each requirement in a view. Interestingly enough, much of DOORS itself is written in DXL. With this architecture, the sky's the limit with regard to how DOORS can be customized—assuming you can grab a competent developer's time, as DXL coding is probably beyond the technical abilities of most project managers.

Managing Change

All right, we have links and views—now what happens when the inevitable occurs and requirements are changed? DOORS flags all incoming or outgoing links that have changed recently. These suspect links can be added to your view using a wizard, so you can quickly examine which objects have changed and what the changes are. Once you've examined the change, DOORS lets you clear the suspicion so it doesn't show up again. However, the process would benefit from automatic change-flagging instead of the user having to regenerate the list for every module.

Milestones also change, and DOORS lets you create baselines for each milestone to view all modifications, additions and deletions between versions. The baselining process is simple, and allows you to create a read-only copy of a module and its history. To further enhance audit tracking, each baseline can be electronically signed by an authorized party—note, this is not a true digital signature (à la RSA).

DOORS distinguishes itself from its peers by adding a time-based component to links. This enables DOORS to support parallel development by teams working on different increments, allowing links to be made to different sets of baselined data.

Rather than waiting for a complete delivery to be finalized before individual components can be frozen, baseline sets can be built up incrementally by adding each new document as it reaches the appropriate milestone. This provides not only an understanding of what requirements looked like in the past, but also what relationships they had with each other at crucial milestones.

Rounding out DOORS' change management tools is an integrated change proposal system that enforces rules about who can change what. The process requires that all requirement modifications be filed as change requests. A change control board can convene, examine each proposal and decide an appropriate disposition for the request. This approach, which introduces another layer of formality into the development process, is especially suited for larger projects with many stakeholders.

Report Card

Overall, I was impressed with DOORS 7.0. It can easily streamline an organization's product development process—DOORS users can capture, manage and report on requirements with more accuracy and less effort than with traditional methods. The product manages to accomplish this without imposing artificial constraints on the project management or development process—the level of formality that DOORS adds to your workflow is largely controlled by how you use it.

Although DOORS can most certainly be used by smaller projects with agile teams, it's best suited for larger projects that involve many stakeholders and reams of requirements—environments where confusion and paperwork typically reign. The decision to adopt DOORS will largely be driven by the complexity of the project underway and whether an organization has the resources to pay its nontrivial price tag. ■