

SpecTRM Product Documentation

Grady Lee, [Safeware Engineering Corporation](#)

What Is It

SpecTRM is a system development environment with particular emphasis on mission-critical and safety-critical systems. SpecTRM facilitates construction of specifications that facilitate development, operations, and maintenance throughout the life cycle of a project. A unique feature of SpecTRM is an emphasis on the construction of software requirements models that can be easily read, reviewed, simulated, and analyzed.

Features

SpecTRM provides a user-friendly editing environment for recording system-level requirements, design and blackbox software requirements models. SpecTRM works on documents called intent specifications. An intent specification is an improved format for writing system and software specifications. Intent specifications do not require additional effort; any good specification already has the same information developed when working with an intent specification. The difference is in how the information is organized. Traditional specifications are abstracted in layers of “what” and “how.” Higher levels of the specification describe *what* is to be done. Levels below describe *how* the system will accomplish its tasks. In addition to describing what and how, intent specifications also record *why* decisions made at lower levels were made the way they were. Often, this intent information is the most pivotal to ensuring a successful system and the most difficult to recover from traditional specification formats. Although SpecTRM begins new projects with a default skeleton specification document, every project is different, and SpecTRM makes it easy to add, remove, change, or relocate sections to fit the needs of individual projects.

Hyperlinks in SpecTRM ensure traceability of safety information and design rationale from the system level through component development. Traceability links from earlier levels of an intent specification make it easier to determine why decisions were made. Reviewers can ensure that important system-level properties have been enforced in the requirements for components, such as software. When the system is changed or components are re-used, traceability links reduce the difficulty and expense of ensuring that the system will continue to perform as desired.

A key feature of SpecTRM’s intent specification is the blackbox requirements model. Almost all software-related accidents, both those that resulted in human injury and those that caused a loss of mission, trace back to flaws in the requirements. Using SpecTRM-RL (SpecTRM Requirements Language) engineers develop requirements models describing the behavior of the software. The modeling language is based on the rigorous mathematics of state machines, allowing the models to be executed and analyzed. Despite the underlying mathematics, the modeling language was designed with readability as a priority; system engineers, software engineers, and domain experts such as pilots, air traffic controllers, and office managers have all been taught to read SpecTRM-RL requirements models with less than a half hour of training.

Over 60 criteria for completeness in safety-critical specifications have been built into SpecTRM-RL's syntax. These criteria have been identified through research, industrial practice, and accident investigation as frequent causes of accidents, including loss of mission. By building them into the syntax of the language, the completeness criteria are applied as the requirements for the software are being developed, instead of during a later review, when responding would impact cost and schedule. A few criteria could not be built into the syntax, and SpecTRM provides automated analysis to check requirements models for two of those.

SpecTRM's capabilities also allow for the execution of the requirements models. Executing the requirements specification allows analysts to find specification errors early in the development process when they have less impact on cost and schedule.

Benefits

SpecTRM assists in finding errors at the requirements level, where resolving errors has less cost and schedule impact. Requirements errors become more expensive to fix the further along they go in the project life cycle. A requirements error caught in system test may cost as much as 1000 times more to fix as it would if caught during the requirements phase of the project. Much of this savings comes in the form of reduced development time. Fixing an error in the requirements specification is far less time consuming than having to rework a design, let alone changing code and associated tests.

Using SpecTRM, engineers trace requirements information, specify design rationale, and update safety information throughout the system life cycle. In safety-critical and mission-critical systems, often the most important information is why a decision was made the way it was. There may be assumptions that the software relies on, or assumptions about the software's behavior made by the rest of the system. Recording this rationale prevents later changes from leading to an accident or the loss of a mission – changes can be evaluated for their impact on the existing design.

SpecTRM helps engineers build desired properties into a system from the beginning, including safety. Many processes emphasize downstream assessment. However, safety information is presented in the form of a design critique is often rationalized away. Risks are dismissed as unlikely because it is too late and too expensive to change the system or software design. SpecTRM instead focuses on building the right system and software from the beginning.

The requirements documentation produced in SpecTRM also acts as a bridge between disciplines. There is a significant cultural gap between engineering specialties, particularly system engineering and software engineering. SpecTRM-RL requirements models are easily readable by system engineers, software engineers, safety engineers, and domain experts such as pilots, security experts, and managers. The enhanced communication between team members reduces the incidences of misunderstandings and conflicting assumptions, again leading to a reduction in cost and development time.

Successes

SpecTRM was developed from specification and analysis techniques that were used to develop the official FAA TCAS II specification. Over the past 10 years, all changes to TCAS have been validated using those specification and analysis techniques. SpecTRM has been adopted and used extensively by the Japan Manned Space Systems Corporation. Safeware has used SpecTRM to develop and analyze requirements models for systems in the automotive, aerospace, and medical device industries, including an electric steering system for Delphi Automotive. Universities in the United State and Sweden have used SpecTRM's extension API as a platform for conducting further research in requirements and safety analysis.

Context in which it is best used

SpecTRM is intended for use in specifying and analyzing requirements for software-intensive mission-critical and safety-critical systems. SpecTRM can be used for any such system, but software with complicated decision-making algorithms is best. Systems with complicated state and mode transition logic are good candidates. These systems will benefit more than systems where the complexity is primarily in numerical calculations.

Compare with Alternative Known Products or Technologies

SpecTRM differs from UML modeling tools in that SpecTRM models focus on black box requirements. The design information included in a UML model obscures the requirements information that is pivotal to system and safety engineering. SpecTRM differs from SCR in its emphasis on readability and review by domain experts with only a little training.

What will a successful collaboration look like?

What will the technology provider do?

Safeware will work with you to develop a proposal planning your collaboration. We will provide a 3-day training course at your site and telephone and email support over the course of the collaboration.

What should the development team do?

Prior to the collaboration, the NASA software development team is invited to communicate with us to determine whether its application is a good fit for SpecTRM. During the collaboration; take the training course; model the software requirements in SpecTRM; and use SpecTRM to validate the requirements.

How will the technology provider work together with the development team to ensure a successful collaboration?

During the proposal process, we will ensure a good match between the project goals and SpecTRM's capabilities. In addition to providing email and phone support to the development team, we will proactively contact the Collaboration PoC, obtain feedback, offer suggestions on effectively employing SpecTRM, and follow up to ensure that we are achieving our collaboration goals through SpecTRM's support of the project's goals.