Using Visualization to Understand Dependability: A Tool Support for Requirements Analysis

Paolo Donzelli[1], Daniel Hirschbach[1], Victor Basili[1,2]

{donzelli, danielh, basili}@cs.umd.edu
[1]*Computer Science Department - University of Maryland*
College Park, 20742 MD, USA
[2]*Fraunhofer Center for Experimental Software Engineering,*
College Park, 20742 MD, USA

Abstract

Dealing with dependability requirements is a complex task for stakeholders and analysts as many different aspects of a system must be taken into account at the same time: services characteristics and quality properties, failure modes and tolerable failure rates, reactions and recovery time in case of failure, and so on. Visualization helps cope with this complexity. In this paper, we build upon a practical framework for eliciting and modeling dependability requirements to show how graphical data representation can facilitate requirements analysis during the requirements elicitation and definition process. An Air Traffic Control System, adopted as a Testbed within the NASA High Dependability Computing Project, is used as a case study.

1. Introduction

The International Federation for Information Processing WG-10.4 [1] defines dependability as *the trustworthiness of a computing system that allows reliance to be justifiably placed on the services it delivers*. "Reliance" is contextually subjective and depends on the particular stakeholders' needs. In different circumstances, stakeholders will focus on different system's properties [2,3], e.g., availability, real-time response, ability to avoid catastrophic failures, and prevention of deliberate intrusions, as well as different levels of adherence to such properties [4,5].

Dependability requirements cover many different aspects of a system at the same time [6,7]. For example, dependability requirements need to address the following: characteristics and quality properties of the most critical services, failures modes and acceptable failure rates, potential hazards, recovery time and system reaction to specific failures, external events that could damage or prevent the system from functioning correctly, and so on.

Dealing with dependability requirements is a complex task for both stakeholders and analysts. Visualization can help reduce this complexity. A graphical representation has the advantage of letting stakeholders and analysts grasp aspects of dependability requirements more immediately and more accurately than textual descriptions.

The strength of visual formalism for human understanding and problem solving is largely recognized in software engineering, and visualization is commonly adopted for program comprehension at code and design [8,9] levels, sometimes at the specification level [10], but rarely at the requirements level. The textual nature of requirements usually prevents the adoption of graphical representation techniques.

In this paper, we build upon a practical framework for eliciting and modeling dependability requirements, i.e., the Unified Model of Dependability (UMD) [11]. UMD is based on a modeling language that adopts a small set of basic dependability concepts to facilitate stakeholders in identifying and precisely formulating their needs. The resulting requirements are clearly structured and suitable for graphical data analysis.

We show how visualization benefits requirements analysis throughout the requirements definition process, in particular during elicitation, early validation and negotiation. During elicitation and early validation, a stakeholder can visualize the impact of his choices, and more easily confirm (validate) or refine his initial requirements, while analysts can identify and highlight potential areas of improvement. During negotiation, stakeholders can better understand each other's position and become more willing to negotiate their initial positions, while

Proceedings of the 2005 29th Annual IEEE/NASA Software Engineering Workshop (SEW'05)
0-7695-2306-4/05 $20.00 © 2005 IEEE

analysts can more easily identify discrepancies and suggest reconciliation solutions.

This work is part of the High Dependability Computing Project (HDCP), a five-year cooperative research agreement between NASA and various universities and research centers to increase NASA's ability to engineer highly dependable software systems. This paper is organized as follows: Section 2 briefly describes UMD, highlighting its main characteristics. Section 3 introduces the case study, an Air Traffic Control System, adopted as a testbed within HDCP [12]. Then, section 4 briefly illustrates how UMD is used to elicit and model requirements, while section 5 provides examples of how visual representation has been used for requirements analysis during early validation and negotiation. Finally, Section 6 concludes and provides an outline of future work.

2. The Unified Model of Dependability

Both stakeholder-oriented and issue-centered, the Unified Model of Dependability is a requirements engineering framework for eliciting and modeling dependability requirements.

UMD is issue-centered as it permits stakeholders to express their requirements by specifying what they see as the actual dependability issue (failure and/or hazard), or class of issues, that should not affect the system or a specific service (scope). For an issue, stakeholders may also specify the tolerable manifestations (measure) and the desired corresponding system reaction. Stakeholders may also specify external events that could be harmful for the system. As illustrated in Figure 1, scope, issue, measure, reaction, and event are the basic modeling concepts of UMD that stakeholders use to express their dependability requirements. For example, for an on-line bookstore system, a requirement expressed using UMD could be: "The book search service (scope) should not have a response time greater than 10 seconds (issue) more often than 1% of the cases (measure); if the failure occurs, the system should warn the user and recover full service in one hour."

UMD is also stakeholder-oriented, as scope, issue, measure, reaction, and event are basic concepts that stakeholders can easily grasp and associate with entities proper to their application domain. Rather than dealing with abstract entities (dependability and its attributes), stakeholders can organize their thoughts about dependability by focusing on practical concepts, enabling them to more effectively map their dependability needs to the context. Moreover, to better support stakeholders in formulating their requirements, and to better address the needs of a particular application context, UMD permits users to refine its basic modeling concepts (Figure 1). For example, to help stakeholders identify failures that should not affect the system or a service, we may suggest the different types of failures that could occur (e.g., response time failures, accuracy failures), but to allow for the specification of more precise requirements, we may also introduce different levels of severity and impact on availability (e.g., stopping and non-stopping failures). Similarly, we can allow stakeholders to select the measurement model most suitable to express the tolerable manifestation of an issue (e.g., Mean Time Between Failure), as well as use different reaction types. For example, stakeholders could express the reaction of the system to a specific issue in terms of warning services (to make the user aware of the situation), mitigation services (to reduce the impact of the failure on the user), alternative services (to provide alternative means to perform the same activity), and recovery behavior (the time and the actions necessary to recover from the failure). Finally, different event types can be suggested to facilitate stakeholders in recognizing external situations that could harm the system (e.g., attacks and adverse conditions).

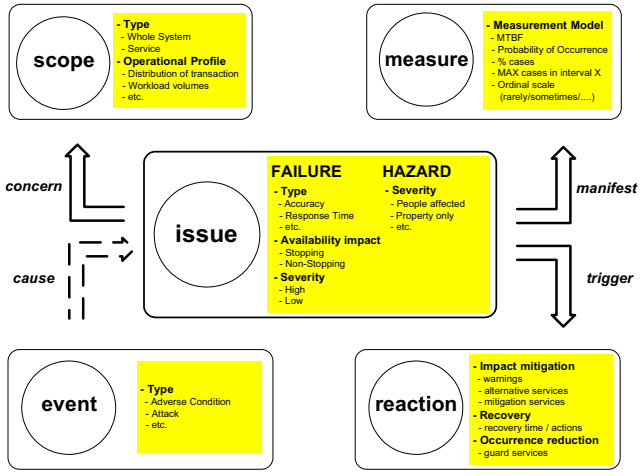

Figure 1: The UMD concepts

To implement the UMD, we developed a web-based tool [11], organized around two main tables:

The Table "Scope" (Figure 3), which allows stakeholders to describe all the services of the system for which dependability could be of concern.

The Table "Issue" (Figure 4), which allows users to specify their dependability needs by defining, for the whole system or a specific service (selected from the scope table), the potential issues (failures and/or

hazards), their tolerable manifestations, the possible triggering external events, and the desired reactions.

3. The Case Study: Applying UMD

Within HDCP, we used UMD to define the dependability requirements of the Tactical Separation Assisted Flight Environment (TSAFE) Testbed, developed at the Fraunhofer Center Maryland [13]. TSAFE [14] is a software system designed to aid air traffic controllers in detecting and resolving short-term conflicts between aircraft. The Testbed has been derived from the TSAFE version developed in [15].

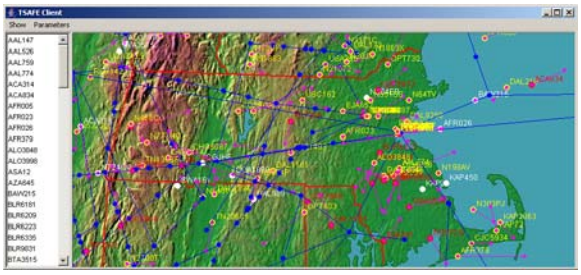

Figure 2 – The TSAFE display

TSAFE provides the air traffic controller with a graphical representation of the conditions (position, planned route, forecasted synthesized route) and the status (conformance or non-conformance to a planned route) of selected flights within a defined geographical area. Its aim is to detect conflicts somewhere between 3 and 7 minutes in the future and issue avoidance maneuvers accordingly. Figure 2 provides a snapshot of the TSAFE display. For TSAFE, a set of functional requirements defining the system was already available. However, there were no precisely stated dependability requirements [15].

During the case study, a small group of computer science researchers and students acted as stakeholders (specifically as air traffic controllers), after being given a short introduction to TSAFE and its purposes, while another acted as an analyst.

4. Requirements Elicitation and Modeling

UMD has been applied in two main steps, scope definition and requirements elicitation and modeling.

Scope definition: All stakeholders, working together and supported by the analyst, have selected from the functional requirements available for TSAFE [15] the services for which they believed dependability could be relevant. The identified services are described in the scope table (Figure 3).

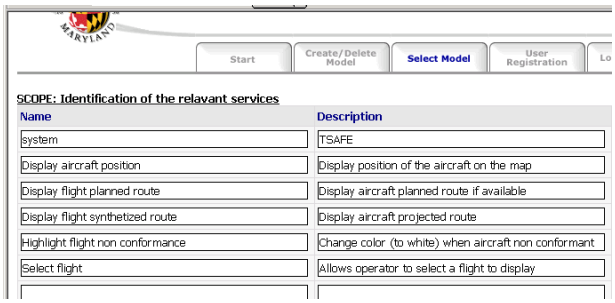

Figure 3: The UMD Tool "Scope" table

Table 1. UMD customization for TSAFE

Failure characterization: Failure Type: • Functional correctness: System or service does not work or it does not implement the functional requirements. • Throughput: Average or peak number of items (aircraft, routes, etc.) per unit of time dealt with by the system or service is less than expected. • Response time: Response time of the system or the service greater than expected. • Peak load: Max number of items handled by the system or the service is less than expected. • Accuracy: The accuracy (Lateral, Longitudinal, Vertical) of the aircraft position or trajectory is less then expected. • Data freshness: The frequency of data updating is less than expected. Failure Impact over Availability: • Stopping: Failure makes the system or service unfit for use. • Non-Stopping: Failure does not make the system or service unfit for use Failure Severity: • High severity: Failure has a major impact on the utility of the system for the operator. • Low severity: Failure has a minor impact on the utility of the system for the operator.	**Hazard characterization**: Hazard Severity: • Catastrophic: Risk of total aircraft destruction. • Severe: Risk of serious damage to the aircraft, serious emergency situation, loss of human lives possible. • Major: Risk of emergency situation, high stress on cockpit crew. **Event characterization**: Event Type: • Environment misbehavior: Any accidental environmental situation that could affect the system. • Radar misbehavior: Any radar anomaly that could affect the system. • Controller misbehavior: Any unexpected controller behavior or action that could affect the system. **Measure characterization:** Measurement Model: • Mean Time Between Failures (MTBF) **Reaction characterization**: Service Type: • Warning Services: Warn user about the situation. • Alternative Services: Provide alternative ways to perform same tasks. • Mitigation Services: Reduce issue impact on the user. • Guard Services: Reduce probability of occurrence of the issue. Recovery Behavior: • Mean Time To Recover (MTTR) – Max Time To Recover (MaxTTR)

Requirements elicitation and modeling: Each stakeholder, supported by the analyst and guided by the structure provided by the tool, has filled as many tables as necessary to define her/his dependability needs (Figure 4). The characterizations of the UMD concepts of scope, issue, event, measure, and reaction have provided useful guidance to stakeholders. Each stakeholder has used the characterizations already available (introduced by the analyst at the beginning of the project or by other stakeholders using the tool earlier), or, whenever necessary, has extended it with his/her own definitions. The characterizations used for TSAFE are described in Table 1. As example of the requirements collected with UMD, we describe one of the tables filled by the stakeholders. Figure 4 illustrates an example of an issue not related to an external event. The stakeholder signals a potential failure for the service "display flight synthesized route," when the response time is greater than 500 ms. This is a Response Time, Non-Stopping, High Severity failure, given the high impact on the service's utility for the operator. For the stakeholder, this failure is also a hazard (Major Hazard), given that he thinks he could miss spotting a plane on a dangerous path. This could lead to an emergency situation and possibly cause high stress on the cockpit crew, required to perform sudden escape maneuvers by the very short-term conflict avoidance systems. The stakeholder identifies this failure as a highly critical one, leading the analyst to suggest MTBF of 2.0E4 (between the values suggested for very high and mission critical availability in table 1). In order to be more confident in the system, the stakeholder asks to introduce a warning service that will advise in case computational time becomes greater than 500 ms, alerting him when greater attention is needed. Finally, the stakeholder asks for the recovery to be performed within one hour by a technician. If this failure condition lasts more than one hour, he feels he would be unable to properly perform his duties, due to the need to maintain a higher than usual level of attention.

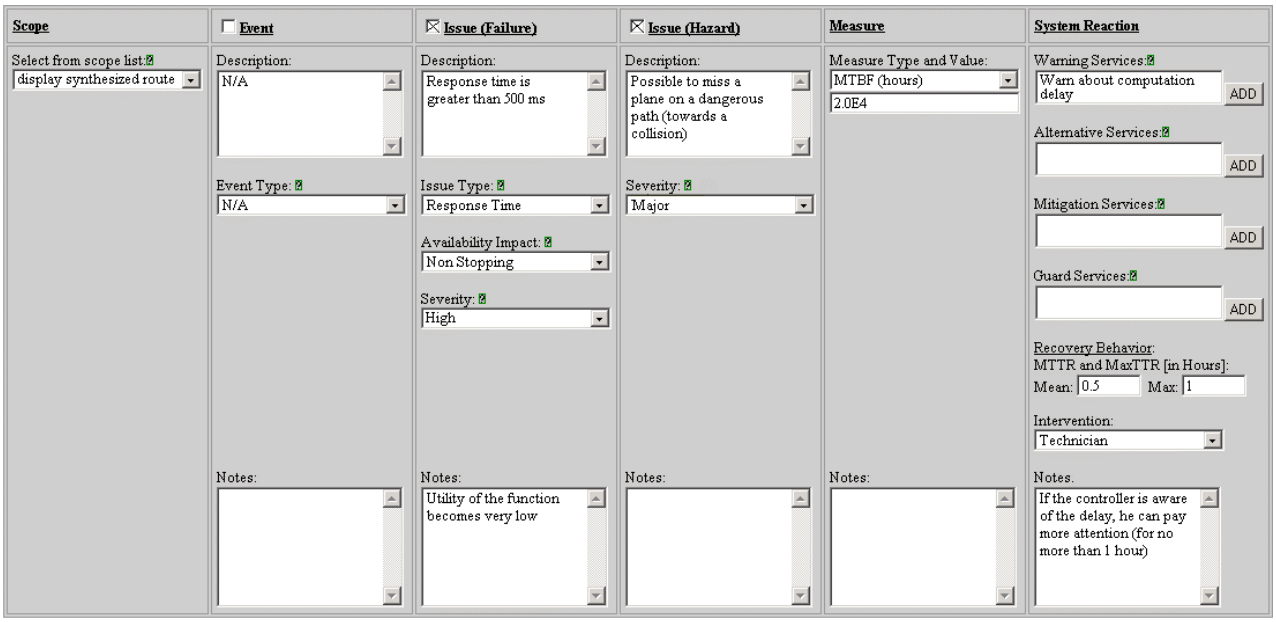

Figure 4: UMD Tool - issue not related to an external event

5. Requirements Analysis through Visualization

The UMD tool allows for analysis of the requirements expressed by the stakeholders at two different levels, in terms of high-level geometrical characteristics of the sets of requirements, and in terms of the emerging system dependability properties. These capabilities are provided through two added-on components.

The **Visual Query Interface (VQI) tool**, developed at the Fraunhofer Center Maryland and based on the idea of the Starfield display [16], is used to analyze requirements geometrically, i.e. to analyze characteristics of sets of requirements rather than individual requirements. VQI allows the spatial visualization of the requirements distributed according the values of two or more of their attributes (e.g., failure type, availability impact and severity; hazard severity; type of external event, type of reaction). Different symbols, colors, labels, and sizes can be used to highlight the attributes of interest.

A prototype tool, developed by combining features provided by *MS Excel* and *Matlab* [17], is used to visually represent the emerging system dependability properties. The measures expressing the tolerable manifestation for each of the identified issues are combined to provide "aggregate values of dependability": for example, the aggregate MTBF of all the failures, or the MTBF of all the failures that are also stopping failures.

In the following, we present several examples of how these features of the UMD Tool have been used for requirements analysis during elicitation and definition of the TSAFE requirements.

5.1 Visualizing geometrical characteristics

The visualization of the high-level geometrical characteristics of the collected sets of requirements is a powerful means that stakeholders and analysts can use to quickly spot potential pitfalls and identify their possible solutions. In the following, we provide two examples to illustrate how this graphical capability of the tool has allowed analysts and stakeholders to identify missing requirements concerning possible failures and neglected external events.

Example 1 - Spatial requirements distribution on the plane "Services vs. Failure Types". Figure 5 illustrates a portion of the UMD Tool display showing how requirements are distributed around the services (y-axis) according their failure type (x-axis). Two colors are used to differentiate high severity (black) from low severity failures (gray), while labels are used to mark stopping failures. By looking at this diagram, the analyst could easily focus on those areas where requirements seemed to be missing and ask stakeholders for clarification. For example, the requirements distribution in Figure 5 (see the dotted circle) showed that a particular type of failure (data freshness) had not been taken into account for a specific service (select flight). Asked whether or not he confirmed such a choice, the stakeholder realized he missed it, so he decided to introduce another requirement.

Similarly, the analyst realized that while there were non-stopping failures of both high and low severity, there was only one failure that the stakeholder had indicated as a stopping and a low severity one (see double circle) Also in this case, the analyst asked the stakeholder to confirm/revise his choice.

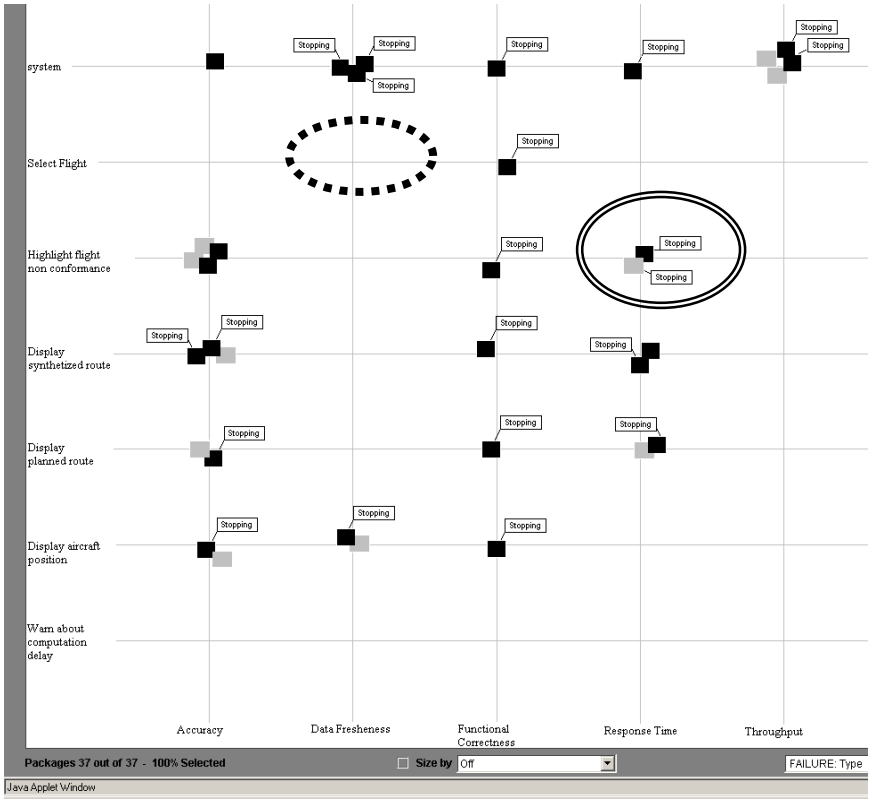

Figure 5: Spatial Requirements Distribution (Services vs. Failure Types)

Example 2 - Spatial requirements distribution on the plane "Services vs. Event Types." In this case (Figure 6), the UMD Tool display shows how the requirements are distributed around the services (y-axis) according their event type (x-axis). Again, two colors are used to differentiate high severity (black) from low severity failures (gray), while labels are used to mark stopping failures. By looking at the diagram, the analyst could easily recognize that the stakeholder had specified various requirements to take into account potentially harmful external events generated by the environment or the controllers, but he had not specified any requirement concerning possible radar misbehavior (dotted circle). Asked to clarify this situation, the stakeholder has decided to add another requirement to specify how TSAFE should behave if the radar transmits inaccurate data.

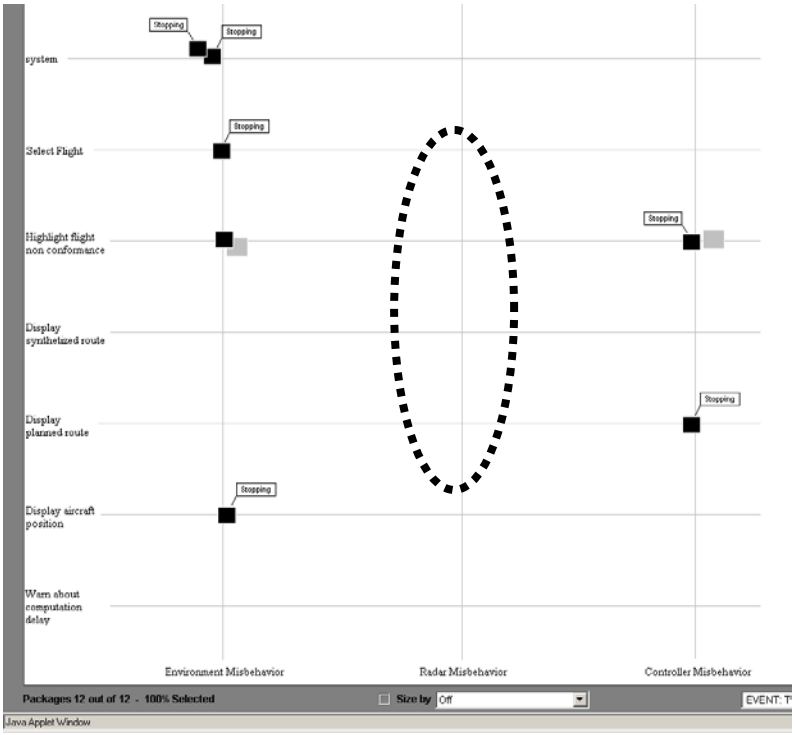

Figure 6: Spatial Requirements Distribution (Services vs. Event Types)

It is important to note that the diagrams produced through VQI can also play a significant role during the negotiation phase, when the analyst has to reconcile the needs emerging from the different stakeholders. In particular, the analysts can quickly compare the sets of requirements produced by different stakeholders to identify the particular areas where negotiation should focus. When the stakeholders have expressed requirements concerning different services and/or failure types, the requirements can be easily merged without negotiation. On the contrary, when they have expressed requirements concerning the same services and failure types, more attention is necessary as discrepancies could be hidden and some negotiation could be needed to reconcile different positions. Stakeholders, for example, could ask for completely different system reactions to the same failures.

5.2 Visualizing "emerging dependability"

While defining requirements, stakeholders necessarily must focus on each single requirement, dealing with a little element of dependability at the time. This situation does include the risk that stakeholders will lose sight of the global result. In other words, although each single requirement could appear to be correct, the emerging system dependability properties, i.e. the dependability properties produced by the combined effect of all the requirements, may actually differ from what stakeholders really want.

The UMD Tool's capability of visualizing system properties unveils these problems, enabling stakeholders to refine their choices very early during the elicitation and negotiation activities. In the following we present some examples of how these

capabilities have been used during the TSAFE case study.

Example 1 – Emerging MTBF. By specifying the dependability requirements for TSAFE using UMD, stakeholders have identified various failures for the different services and for each they have defined a tolerable manifestation, i.e. MTBF. The MTBF specified for each requirement may be aggregated to compute the resulting MTBF for the system or a specific service. For example, the formula for computing the MTBF for a service is the following:

$$MTBF(service) = \frac{1}{\sum_{i=1}^{N} \frac{1}{MTBF(failure_i)}},$$

where $failure_i$ is any failure specified for the service.

As an example, Figure 7 illustrates the MTBF computed for each service on the basis of the stakeholder-specified requirements, separating MTBF for high severity failures and MTBF for low severity failures.

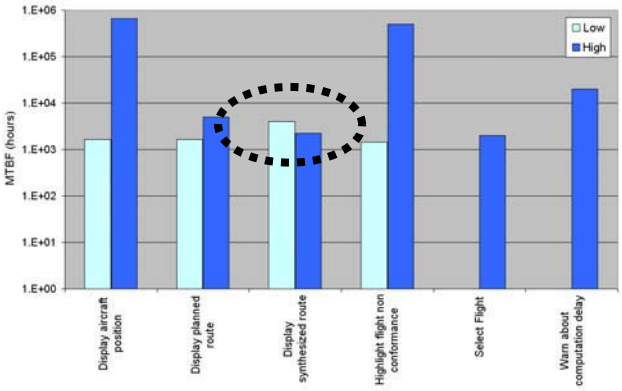

Figure 7: MTBF for TSAFE services (distinguished by failure severity)

By looking at this diagram (Figure 7), the analysts could easily spot an unnatural situation concerning the service "Display synthesized route": The emerging MTBF for low severity failures is, in fact, higher than the MTBF for high severity failures (see dotted circle). This situation can be deemed unnatural, as it would be more expected for the stakeholder to require higher MTBF for high severity failures than the other way around. In order to better understand the problem, the analyst decided to consider all the failures that the stakeholder had specified for that service. These (and the corresponding MTBF) are illustrated in Figure 8.

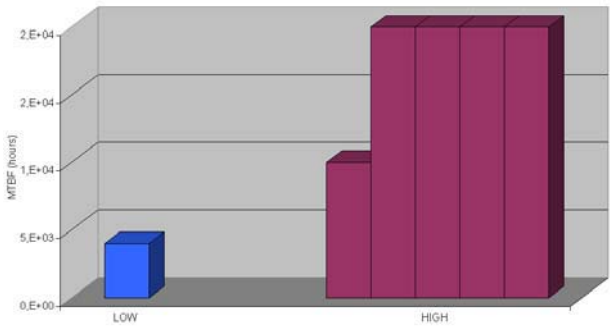

Figure 8: MTBF of the failures specified for the service "Display Synthesized Route"

As this diagram shows, the stakeholder had specified only one requirement concerning a low severity failure (with MTBF of 4000 hours), but five requirements concerning high severity failures, with a MTBF ranging from 10000 to 20000 hours, leading to a combined MTBF of 2222 hours (see Figure 7). Thus, although the stakeholder has correctly required a higher MTBF for the high severity failures, the combined effect of having five different failure modes affecting the same service has produced a lower than expected MTBF. After identifying the problem, the analyst had asked the stakeholder to confirm/revise his initial choices. As result, the stakeholder assigned higher MTBF values for the high severity failures.

Example 2 – Emerging availability. As we aggregate the values of MTBF specified by the stakeholders within the different requirements to compute the emerging MTBF for the system or a service, we can also compute the emerging availability for the system or a service. In particular, by having the MTBF of all the stopping failures affecting a specific service (or the system), and knowing the corresponding desired recovery time, the desired availability for the service (or the system) can be computed as follows:

$$Availability(service) = \frac{MTBF(service)}{MTBF(service) + MAX(MTTR(failure_i))},$$

where $Max(MTTR(failure_i))$ is the maximum of the MTTR specified for all the failures concerning that service.

As an example, Figure 9 illustrates the availability computed for each service on the basis of the requirements specified by a stakeholder. In this case, by looking at the diagram in Figure 9, the analysts noted that the resulting availability for the service "Display synthesized route" was lower than for the other services. Due to the criticality of this service, he decided to clarify the situation with the stakeholder,

resulting in updated values for the MTBF and MTTR specified in the initial requirements.

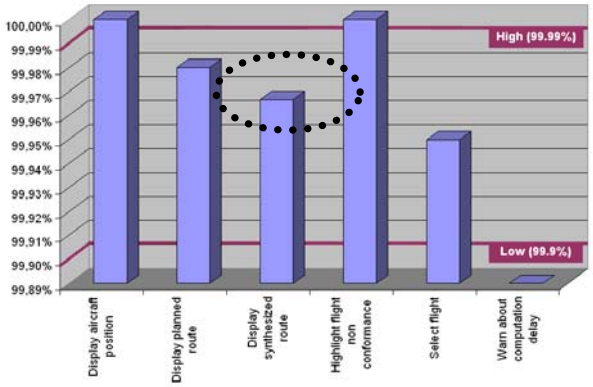

Figure 9 –Availability for TSAFE services

Example 3 - Global view of dependability. The desired values defined for the MTBF by the stakeholders while specifying the requirements can be combined to compute the MTBF for each service and failure type. For example, the results obtained on the basis of the requirements specified by a stakeholder are illustrated in Figure 11. This diagram provides a global view of the system dependability desired by the stakeholder: For each service it clearly shows the required MTBF concerning its functional behavior (functional correctness) and each quality characteristic (accuracy, response time, throughput, etc.).

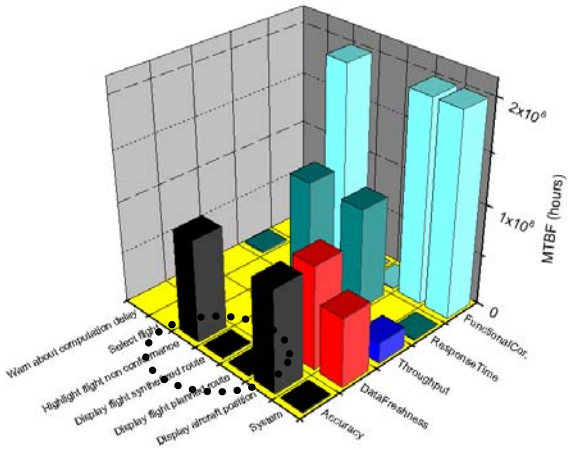

Figure 11: Global view of TSAFE Dependability

The analyst can use the diagram in Figure 11 to further assess the requirements expressed by a stakeholder (for example, the dotted circle highlights potential neglected areas), but also to quickly compare the requirements produced by different stakeholders before proceeding with the negotiation phase. The diagram allows the analyst to distinguish areas of potential agreements (e.g., where stakeholders focused on different services and quality characteristics, or expressed similar values of MTBF for the same services and quality characteristics) from areas of possible risk (e.g., where stakeholders expressed requirements concerning the same services and quality characteristics, but requiring significantly different MTBF values).

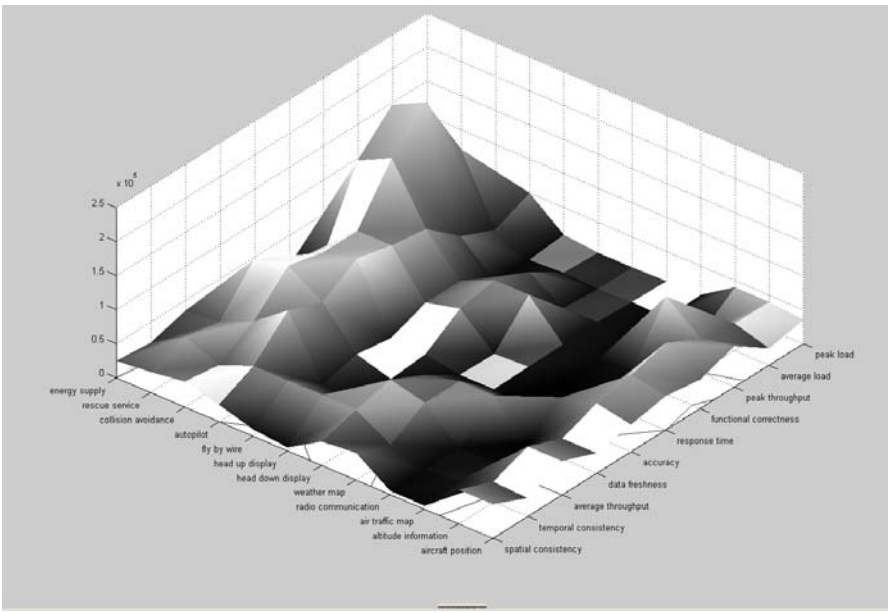

Figure 12: Example of Global view of Dependability using

Finally, it is worth noting that in case of systems more complex than TSAFE, i.e. systems with more services and failure modes, a continuous surface could be adopted to visualize the data. As example, Figure 12 shows the MTBF for each service and failure type obtained using dependability requirements expressed by a single stakeholder for a basic aircraft avionics system, including services such as autopilot, fly by wire, collision avoidance, and so on. The surface provides an intuitive way to convey stakeholder's dependability concerns. In Figure 12, for example, we see that higher MTBF values are localized around the most critical (at least for the stakeholder) services provided by the avionics system (e.g. collision avoidance, rescue service). Similarly, we can see the failure types of which the stakeholder is most afraid (e.g., functional correctness and response time), as we can see the relatively less relevant areas, for example the air traffic map.

6. Conclusions

Dealing with dependability requirements is a complex task for both stakeholders and analysts. In this paper, we have built upon a practical framework for eliciting and modeling dependability requirements (the Unified Model of Dependability) to show how requirements visualization can play an important role to facilitate analysis during the requirements elicitation and definition process. UMD is based on a modeling language that adopts a small set of basic dependability concepts (scope, issue, measure, reaction, and event) to facilitate stakeholders to identify and precisely formulate their needs. The resulting requirements are clearly structured and suitable for graphical data analysis.

An Air Traffic Control System, adopted as a Testbed within the NASA High Dependability Computing Project, was used as a case study. The clarity of the UMD modeling language and the graphical analysis capabilities provided by the UMD Tool have been key factors during the elicitation and definition of the requirements for TSAFE. On the one hand, stakeholders could easily express themselves (elicitation), visualize the impact of their initial choices and act consequently by confirming or refining them (early validation), and understand each other's positions while reconciling their needs with those held by the other stakeholders (negotiation). On the other hand, the analyst facilitated in spotting potential areas of risks with the requirements expressed by a stakeholder, for which it was necessary to require clarifications or suggest solutions (elicitation and early validation), but also in understanding discrepancies among stakeholders while trying to combine and reconcile their different needs (negotiation).

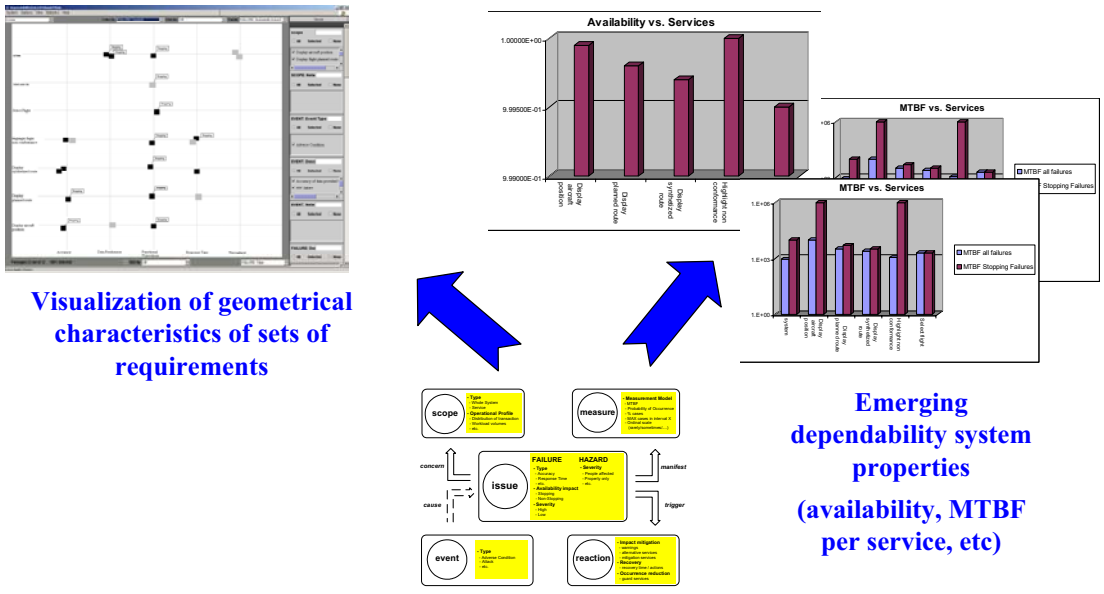

Figure 13: UMD's support to requirements analysis during validation and negotiation

The case study results have increased our confidence in UMD's ability to support not only the requirements elicitation, but also validation and negotiation (Figure 13). In particular, during the case study, UMD' s capabilities allowed us to identify and correct various requirements omissions and discrepancies. Future work will address the possibility of improving automatic support by encapsulating into the tool the adopted "rules of thumb" to draw analysts' attention on omissions (e.g., areas of the requirements space that have not been covered by stakeholders), anomalies (e.g., stakeholders using unusual severity classification for certain types of failures), and potential discrepancies (e.g., stakeholders specifying failure modes concerning the same system functionalities but requiring different system reactions).

7. References

[1] International Federation for Information Processing (IFIP WG-10.4), www.dependability.org.

[2] Laprie Jean-Claude, Dependability: Basic Concepts and Terminology, Dependable Computing and Fault Tolerance, Vienna, Austria, Springer-Verlag, 1992.

[3] Randel B., Dependability, A unifying concepts, Proceedings of Computer Security, Dependability and Assurance: from needs to solutions, York, UK & Williamsburg, VA, USA, July & November 1998.

[4] Boehm B., Huang L., Jain A., Madachy R., The ROI of Software Dependability: The iDave Model, IEEE Software, Vol. 21, Issue 3, May/June, 2004.

[5] Mary Shaw, Everyday Dependability for Everyday Needs, Supplemental Proc of 13th International Symposium on Software Reliability Engineering, Maryland, 2002.

[6] Melhart Bonnie, Stephanie White, Issues in defining, analyzing, refining, and specifying system dependability requirements, IEEE Conference on the Engineering of Computer Based Systems, April 2000.

[7] Virtanen S., Reliability in Product Design – Specification of Dependability Requirements, IEEE Reliability and Maintainability Symposium, 1998.

[8] Ball T., Eick S.G., Software visualization in the large, IEEE Computer, April 1996.

[9] Zernik D., Snir M., Malki D., Using visualization tools to understand concurrency, IEEE Software, may 1992.

[10] Kim S., Carrington D., Visualization of formal specifications, IEEE, 1999.

[11] Basili, V., Donzelli, P., Asgari, S. The Unified Model of Dependability: Putting Dependability in Context, IEEE Software, Vol.21, Issue 3, Nov/Dec 2004.

[12] High Dependability Computing Project, http://hdcp.org

[13] Asgari S., Basili S., Costa P. Donzelli P., Hochstein L., Lindvall M., Rus I., Shull F., Tvedt R., Zelkowitz M., Empirical-based Estimation of the Effect on Software Dependability of a Technique for Architecture Conformance Verification, ICSE 2004 Workshop on Architecting Dependable Systems (WADS), Edinburgh, Scotland – UK, 25 May, 2004

[14] Erzberger Heinz, The Automated Airspace Concept, The 4th USA/Europe Air Traffic management R&D Seminar, Santa Fe, New Mexico, USA, Dec 3-7, 2001.

[15] Dennis G. TSAFE: Building a Trusted Computing Base for Air Traffic Control Software (MSc Thesis), January 2003.

[16] Ahlberg C. Shneiderman B., Visual information seeking: Tight coupling of dynamic query filters with starfield displays, Proc. CHI'94 Conference: Human Factors in Computing Systems, ACM, New York, NY (1994), 313-321

[17] Matlab, http://www.mathworks.com

Acknowledgements

The authors wish to acknowledge support from the NASA High Dependability Computing Project under cooperative agreement NCC-2-1298.

The authors wish to thank the researchers on the HDCP project team for their insights and suggestions and Jennifer Dix for proof-reading this paper.