Ontology based Specifications for Software Reliability Advancement

Shilpa Sharma
Medi-Caps Institute of Technology & Management
Indore, Madhya Pradesh,India

Maya Ingle
Devi Ahilya University
Indore, Madhya Pradesh, India

ABSTRACT
Presently, achieving highly reliable software from the customer's perspective is a demanding job for all software engineers and reliability engineers. Consequently, reliability may be defined by the user's requirements. In addition, specifications are the basis for software development. Therefore, all means applicable ought to substantiate the specifications concerning requirements. To specifically and succinctly define the reliability, specifications must be written in formal and highly expressive language. Formal ontologies enable use of automated consistency checking of the software system with respect to declarative specifications. Therefore, we have proposed Ontology based protocol (*OntoReliability*) for developing software specifications in support of reliability advancement. We also present examples from the practice of our protocol that constitutes description, preconditions, post conditions, standard course, proxy course, exceptions, inclusions, primacy, rate of uses, exceptional requirements and remarks and concerns. This, conversely, necessitates revealing the benefits of implementing an *OntoReliabilty* protocol.

Keywords
OntoReliability Protocol, OntoRelSpecifications

1. INTRODUCTION
Reliability accord scheduled prior to software development, has been attracting a growing attention among software engineers and reliability experts. Software specification decisions have a direct impact on system aspects such as overheads, time-to-market, and quality [1]. This consideration results in software reliability accord in the phase of software specification development. The reliability of a software system is defined as the probability that a system will perform as per its specifications [2]. All required characteristics of the software to be implemented are determined by specifications. Accordingly, form the starting point of any software development process. Specifications are also an important means of communication between users and developers. As formalized expressions of the requirements, specifications stand at the borderline between informal and formal descriptions of an application area [3]. For an immediate reflection of the consequences of the specifications and for an early substantiation, it has been suggested that specifications should furthermore be accomplishable. Therefore, ontology based specifications has been proposed to serve as prototype that may be used as an evolutionary approach for software development. This is particularly imperative since in many projects the requirements cannot initially be stated completely and precisely. The accomplishment makes ontology based specifications an optimal communication mode between users and developers for the intended system behavior discussion. It is a process that concepts or relationships provided must be endorsed by users, during specification phase in software engineering. We use ontology to express the approved degree that is reliability. According to which wrong knowledge representation can be cut and the approved knowledge by most users can be hold. Ontology is the explicit specialization of conceptualization, which describes domain knowledge by an abstract model, and hence widely applicable in software reliability engineering [4]. Ontology based specifications describe the required behavior of a future software system in problem oriented terms, i.e. the specifications form a conceptual model of the system. These integrates description, preconditions, post conditions, standard course, proxy course, exceptions, inclusions, primacy, rate of uses, exceptional requirements and remarks and concerns.

Section 2 reveals the related work for software reliability advancements using natural languages, formal specification languages declarative languages such as the functional language. Aiming at software reliability problem, a protocol of reliability advancement based on ontology is put forward for building software specifications from five dimensions in Section 3. In Section 4, we present some case studies to verify the influence of *OntoReliability* protocol on building software specifications, which can help to abate the burden of software developers and reliability experts. Finally, we conclude with deepen common understand between users about domain knowledge, and improve the reliability of software to an optimal extent.

2. RELATED WORKS
Conventionally, specifications have been written in natural language, and later more and more specifications have been written in formal specification languages [5]. Compared with specifications in natural language, formal specifications have many advantages [6]. Since a formal language has a well-defined syntax and a well-defined semantics, all details of a specification must be stated explicitly; thus missing, ambiguous, or inconsistent information can much easier be found. In addition, formal reasoning about the specification is possible, specifically verification and validation with respect to the requirements. Declarative languages, such as the functional language or the logic language Prolog [7] [8], state what is to be computed in a form that is largely independent of how the computation is performed. Declarative languages are based on sound mathematical foundations, have well-defined semantics, permit descriptions at a very high level of abstraction, and are referentially transparent. Thus declarative languages are especially suitable as specification languages. Traditionally, logic has been used as a powerful, concise, and declarative language for software specifications. The restriction to Horn clause logic and the mechanization of proofs which lead to logic languages like Prolog makes these specifications executable.
In addition, the operational profile which is a quantitative characterization of how system will be used is thus essential in software reliability engineering. Using an operational profile to guide system testing ensures that if testing is

terminated and the software is shipped because of imperative schedule constraints, the most-used operations will have received the most testing, and reliability level will be the maximum. Experience to date indicates that operational profiles are beneficial even when very simple and approximate. For example, one project used an operational profile defined only on five operations. However, defining operational profiles in range of 50 to 200 operations has usually been definitely worth the extra effort [9].

Also, several proposals have been made to predict reliability at the architecture level. Most promising reliability evaluation methods, i.e. the methods from [10], [11], [12], [13], and [14] have been compared in [15] from the viewpoint of software architecture. The Reliability and Availability Prediction (RAP) method and the Quality Requirements of a software Family (QRF) method [16] [17] highlight how quality requirements have to be defined, represented and transformed to architectural models. However, the existing proposals do not consider the system reliability and architecture modeling systematically. Most of them lack tool support for reliability evaluation. Moreover, it is also common to these methods that traceability of reliability requirements to software architecture is missing.

3. ONTORELIABILITY PROTOCOL

Figure 1 shows the dominant steps that contribute to affect software reliability. First the errors may introduce at the stage of setting up specifications. These errors could be due to incomplete information in the requirements, erroneous information in the requirements or error in translating the requirements into specifications. All these result in specification error. A further independent component of error is added by the developer/programmer in converting the specifications into the program. The erroneous program when executed leads to a system error. Thus, products in which the defects are present at all times but are revealed as malfunction by random inputs. Therefore, we have proposed a protocol based on ontology for reliability advancement

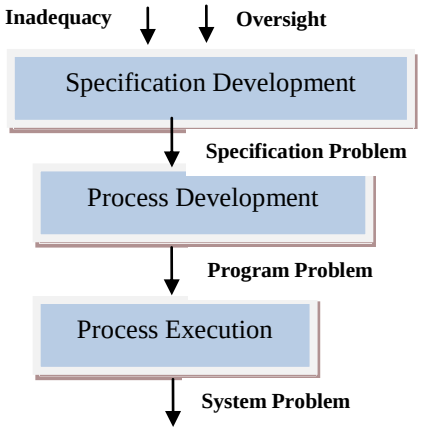

Figure 1: Steps affecting Software Reliability

In order to improve the software reliability an *OntoReliability* Protocol has been proposed for developing software specifications. It improves external world representation in the users' intentions. It controls the changes to the beliefs vigilantly with a meta level structure specifically using ontology for contextual discrimination. The protocol commenced reliability with abet of description, preconditions, post conditions, standard course, proxy courses, exceptions, inclusions, primacy, rate of uses, exceptional requirements and remarks and concerns as illustrated in Figure 2.

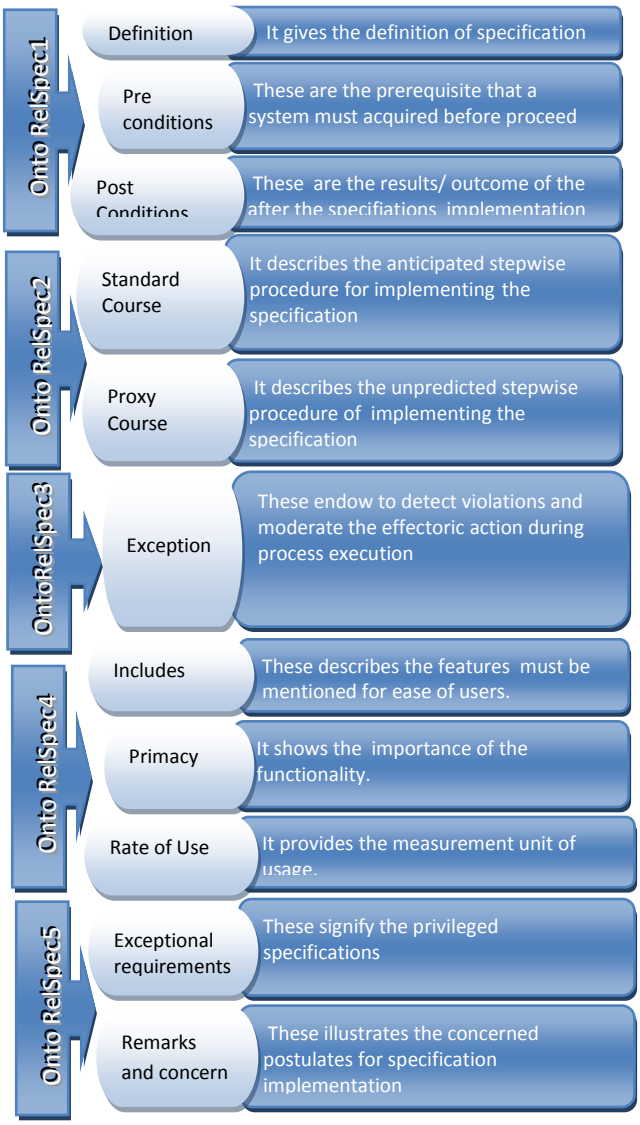

Figure 2: *OntoReliabilty* Protocol

3.1 *OntoRelSpecifications1*

It describes complete knowledge representation with respect to all desires that must be consistent across all domain users. Therefore, *OntoRelSpecifications1* are composed of description, preconditions and post conditions. As illustrated in diagram definition is the characterization of user task. Subsequently, preconditions have been proposed in order to get acquaintance of the existing environment autonomy, constraints and controls. Later, the post conditions are mentioned to envisage the consequences of specification implementation.

3.2 *OntoRelSpecifications2*

It signifies the predicted process execution in advance along with the alternate completing approach, if obligatory. The following attributes are taken into account while designing the standard and proxy courses.

- System does not sap
- System may undergo several updates during the life cycle
- System fixes may introduce new problems
- Software testing is usually incomplete due to its complexity (large number of states)

3.3 *OntoRelSpecifications3*

It illustrates the occurrence of the exception is assumed to be immediately captured and can be automatically stored in exception repository according to different classifications. These anticipated exceptions includes as service unavailability, deadline expiry, external trigger and rationality violation.

3.4 *OntoRelSpecifications4*

The specification specifically focused on inclusions, primacy, and rate of uses. The inclusions define the features that must be mentioned for ease of users such as confining the system decision information, authentication requisitions, and security check services. Subsequently, primacy defines the threshold values such as *High*, *Medium* and *Low*. These values have been set to remove the off beam comprehension on the basis of users' evaluation for particular functionalities. Next, rate of use indicates the unit of measurement for specification usage.

3.5 OntoRelSpecifications5

The specification comprised of exceptional requirements and remarks and concerns. The special requirements specify the privileges given to user in order to endorse the system. In addition, notes and issues have been proposed as reminder to specification developer in order to sustain the trustworthiness.

4. CASE STUDIES

These studies are being conceded using *OntoReliabilty* protocol, to produce the software specifications for different applications of various domains. It includes description, preconditions, post conditions, normal flow, alternative flows, exceptions, inclusions, priority, frequency of uses, special requirements and notes and issues.

4.1 Airline Flight Reservation System (AFRS)

The AFRS is a Web-based application that can accept client requests, list searched results, process booking, payment, modification and cancellation to existing reservations. Users do not have to personally go to the counter or contact airline representatives through the telephone, but only access AFRS through any browser to book their flights. By using AFRS, user can not only save time but also get much larger search space from which they have higher chance to find a suitable air flight. We have developed the specifications for placing reservation, change or cancel reservation and Update/Add/Delete flight information/User reservations using *OntoReliabilty* as shown in Table 1, 2 and 3 respectively.

Table 1: Specifications for placing reservation using *OntoReliability* protocol

OntoRelSpecifications1	Description:	A customer accesses the AFIRS from the Internet, optionally search for specific ticket/flight information of interest, selects ticket(s), and places reservation.
	Preconditions:	None
	Post conditions:	Database of available tickets is updated to reflect items in this order. Remaining tickets number is updated.
OntoRelSpecifications2	Standard course:	**1.0 Order a Single Ticket** • Customer uses the web interface to enter a certain query to view flight information for a specified interest. • System displays available flight information satisfied the query. • Customer selects one or more items from page. Customer can also click on a particular ticket to see the detailed information. • System displays reservation with detailed price information including all taxes. • Customer confirms reservation or requests to modify reservation (back to step 3). • Customer specifies payment method. • Customer indicates that reservation is complete. • System confirms acceptance of the order. • System sends Customer an e-mail confirming order details, price, and additional links to access the ticket details or for potential modification. • System stores order in database, and updates available ticket information (database).
	Proxy course:	**1.1 Order multiple tickets** (branch after step 8) • Customer asks to place another reservation. • Return to step 2. **1.2. Order the Last minute deals** (after step 2) • Customer orders the daily special from the menu. • Return to step 5.
OntoRelSpe	Exceptions:	**1.0.E.1 Concurrent access from multiple users (when there is less available ticket than potential users, demand surpass supply)** (at step 1) • System informs Customer that ticket no longer available. • 2a. Customer cancels the ticket order. • 2b. System terminates. • 3a. Customer requests to select another ticket. • 3b. System restarts. **1.0.E.2 Cutoff time for available ticket (the cutoff time is usually 5 hours before the departure time of the flight)** (at step 1)

cifications3		• System informs Customer that the cutoff time policy occurs. • 1a. System denies the access to the particular ticket information terminates. **1.2.E.1 the user input query is not reasonable (e.g. departure time is behind arrival time)** (at step 1) • System informs Customer of right form of query to input. • Customer changes query.
OntoRelSpecifications4	Includes:	None
	Primacy:	High
	Rate of Use:	Approximately 400 users, average of one usage per day
OntoRelSpecifications5	Exceptional Requirements:	Customer shall be able to cancel the order at any time prior to confirming the order. Customer shall be able to view all tickets he reserved within the previous six months. (Priority = medium)
	Remarks and concerns:	The default time zone of departure/arrival information is the local time zone of specific city. If customer doesn't need to have an account until reservation is placed.

Table 2: Specifications for change/cancel reservation using *OntoReliability* protocol

OntoRelSpecifications1	Description:	Customers who have reservations in AFIRS should be able to modify or cancel these reservations before a certain cutoff time.
	Preconditions:	Customer is logged into AFIRS.
	Post conditions:	Customer has placed certain actions on existing reservations.
OntoRelSpecifications2	Standard course:	**2.0 Reservation modification or cancellation** • Customer requests to change or cancel reservation. • System invokes Authenticate User's Identity. • System verifies Customer's identity and provides the login view menu for customer. • Customer clicks on the reservation section and chooses one of the reservations to modify or cancel. • Customer confirms desire to do modification or cancellation. • System checks the cutoff time and permit the modification/cancellation requested by customer. • System asks Customer to confirm his or her decision. • System sends corresponding update information to the database of ticket/flight information. • System informs Customer the change and provides confirmation number of the transaction.
	Proxy course:	None
OntoRelSpecifications3	Exceptions:	**E.1 Customer identity authentication fails** (at step 2) • System gives user two more opportunities for correct identity authentication. • 2a. If authentication is successful, Customer proceeds. • 2b. If authentication fails after three tries, System notifies Customer, logs invalid authentication attempt, and terminates. **E.2 The cutoff time policy is applied** (at step 6) • System informs Customer that he cannot make the modification/cancellation and explains why. • System terminates.
OntoRelSpecifications4	Includes:	Authenticate User's Identity
	Primacy:	High
	Rate of Use:	Once per user on average
OntoRelSpecifications5	Exceptional Requirements:	User authentication is performed per corporate standards for medium-security applications.
	Remarks and concern:	Expect low frequency of executing this use case. But relatively high frequency during the hot season (Christmas)

Table 3: Specifications for Update/Add/Delete Flight information/User reservations using *OntoReliability* protocol

OntoRelSpecifications1	Description:	The Administrator may modify the flight information and prices for a specified date to reflect changes in availability or prices or to define last minute deal. Administrators can also Update/Add/Delete User Reservations in some cases.
	Preconditions:	Database already exists in the system.
	Post conditions:	Modified database has been saved.
OntoRelSpecifications2	Standard course:	**Update/Add/Delete Flight information/User reservations** • Menu Manager requests to view the menu for specific ticket/flight information. • System displays the menu. • Menu Manager modifies the menu to add new information, remove or change items, create or change deal, or change prices, number of seats available etc. (invoke the database management language module through interface) • Menu Manager requests to save the modified menu. • System saves modified menu. • If the change is about user reservations, send notification to users by e-mail
	Proxy course:	None
OntoRelSpecifications3	Exceptions:	**E.1 No item exists for specified information** (at step 1) • System informs Administrator that no menu exists for the specified date. • System asks Administrator if he would like to add a new item. • 3a. Administrator says yes. • 3b. System invokes Database interface. • 4a. Menu Manager says no. • 4b. System terminates. **E.2 Item specified is the past information** (at step 1) • System informs Administrator that the item requested cannot be modified. • System terminates.
OntoRelSpecifications4	Includes:	None
	Primacy:	High
	Rate of Use:	Approximately 20 times per week by one user
OntoRelSpecifications5	Exceptional Requirements:	The Administrator may cancel out of the modification function at any time. If any item has been changed, the system shall request confirmation of the cancellation.
	Remarks and concern:	If the Administrator is doing modification of certain information, that information should be temporally invisible/inaccessible for customers.

4.2 Web Accessible Alumni Database System (WAADS)

The WAAD encompass numerous files and information from the Alumni Database, as well as files on the department server system. This system will be completely web-based, linking to WAAD and the remote web server from a standard web browser. An Internet connection is necessary to access the system. The WAAD web site will be operated from the departmental server. When an Alum connects to the University Web Server, the University Web Server will pass the Alum to the Departmental Server. The Departmental Server will then interact with the Alumni Database through BDE, allows the Windows type program to transfer data to and from a database. We have designed WAADS to allow alums to fill out a survey form, create a new database entry, update an existing database entry, or contact another alum using *OntoReliabilty* as shown in Table 4, 5, 6, 7 and 8 respectively.

Table 4: Specifications to Access Alumni Home Page using *OntoReliability* protocol

OntoRelSpecifications1	Description:	The Departmental Web Server is waiting on an Alum to connect
	Preconditions:	Alum is connected to the Internet and on the CIS home page
	Post conditions:	The Alum is on the Alumni Home Page
OntoRelSpecifications2	Standard course:	• The Alum connects to the University Web Server. • The Alum selects the Alum link on the CIS home page. • The University Web Server passes the Alum to the Alumni Home Page.
	Proxy course:	• None
OntoRelSpecifications3	Exceptions:	If there is a connection failure the Departmental Server returns to the wait state
OntoRelSpecifications4	Includes:	Alum authentication
	Primacy:	High
	Rate of Use:	Approximately 100 users, average of one usage per day
OntoRelSpecifications5	Exceptional Requirements:	University Web Server sends the Alum to the Departmental Server. The Departmental Server presents the Alum with the Alumni Home Page.
	Remarks and	None

	concern:	

Table 5: Specifications of survey using *OntoReliability* protocol

OntoRelSpecifications1	Description:	The Alum chooses to fill out a survey
	Preconditions:	The Alum is connected to the Internet and on the CIS Alumni Home Page
	Post conditions:	The survey record is created in the Survey Table of the Alumni Database.
OntoRelSpecifications2	Standard course:	• The Departmental Server presents the Alum with a form. • The Alum fills in the form and click submit • The Departmental Server checks to see if all required fields are not empty. • If the required fields are not empty, the Departmental Server creates a new record in then Survey Table of the Alumni Database. • If any of the required fields are empty, the Departmental Server returns a message and returns the Alum to the Survey form. • The Departmental Server returns the Alum to the Alumni Home Page
	Proxy course:	• None
OntoRelSpecifications3	Exceptions:	If the connection is terminated before the form is submitted, the fields are all cleared and the Departmental Server is returned to the wait state.
OntoRelSpecifications4	Includes:	Alum authentication
	Primacy:	High
	Rate of Use:	Approximately 100 users, average of one usage per day
OntoRelSpecifications5	Exceptional Requirements:	None
	Remarks and concern:	None

Table 6: Specifications to create new entry using *OntoReliability* protocol

OntoRelSpecifications1	Description:	The Alum chooses to create a new entry on the Entries page
	Preconditions:	The Alum must be connected to the Internet and on the CIS Entries page.
	Post conditions:	A record is created in the Alumni Table of the Alumni Database.
OntoRelSpecifications2	Standard course:	• The Alum clicks on add a new entry. • The Departmental Server returns a form. • The Alum fills in the form and clicks submit. • The Departmental Server checks to see if any required field is empty. • If any required field is empty the Departmental Server will send a message and return the Alum to the new entry form page. • If no required field is empty the Departmental Server will create a new record in the Alumni Table in the Alumni Database, and return the Alum to the CIS Alumni Home Page. • The Alum may select Cancel. • If the Alum selects Cancel, the form is cleared and the Alum is returned to the CIS Alumni Home page.
	Proxy course:	• None
OntoRelSpecifications3	Exceptions:	• If the connection is terminated before the form is submitted, the fields are cleared and the Departmental Server is returned to the wait state. • If the connection is terminated after the form is submitted, but before the Alum is returned to the CIS Alumni Home Page, the record is created in the Alumni Table of the Alumni Database.
OntoRelSpecifications4	Includes:	None
	Primacy:	High
	Rate of Use:	Approximately 100 users
OntoRelSpecifications5	Exceptional Requirements:	None
	Remarks and concern:	None

Table 7: Specifications to update an entry using *OntoReliability* protocol

OntoRelSpecifications1	Description:	The Alum chooses to update an existing entry in the Alumni Database
	Preconditions:	The Alum must be connected to the Internet and on the CIS Entries Page.
	Post conditions:	The record in the Alumni Table of the Alumni Database has been updated and the

		Alum is returned to the CIS Alumni Home Page.
OntoRelSpecifications2	Standard course:	• The Alum clicks on update an entry link. • The Departmental Server returns a form. • The Alum enters his/her year of graduation. • The Departmental Server queries the Alumni Database for that particular year and returns a table of all graduates from that year in a form with radio buttons and requesting their password. • If the password does not match the Departmental Server returns a message and allows the Alum to try again. • If after 3 tries the password does not match, the Departmental Server will return a message telling the Alum to contact the CIS designated faculty member to receive their password. • If the password matches go to 8. • The Departmental Server returns a form with the data for that Alum in it and a message to update the data they wish and click submit. • The Departmental Server with replaces the old data with the new data and returns the Alum to the CIS Alumni Home Page.
	Proxy course:	• If after three attempts to match the name and password the Departmental Server will return a message and block the Alum from the update section.
OntoRelSpecifications3	Exceptions:	• If the connection is terminated before the form is submitted, the fields are cleared and the Departmental Server is returned to the wait state. • If the connection is terminated after the form is submitted, but before the Alum is returned to the CIS Alumni Home Page, the record in the Alumni Table of the Alumni Database is updated and the Departmental Server is returned to the wait state
OntoRelSpecifications4	Includes:	• None
	Primacy:	• High
	Rate of Use:	• None
OntoRelSpecifications5	Exceptional Requirements:	• None
	Remarks and concern:	• None

Table 8: Specifications for Search for an Alumni/E-mail an Alumni entry using *OntoReliability* protocol

OntoRelSpecifications1	Description:	The Alum chooses to search/e-mail Alum.
	Preconditions:	The Alum is connected to the Internet and on the CIS Alumni Home Page.
	Post conditions:	The Alum receives the information on the requested Alum, receives e-mail confirmation message, or is returned to the CIS Alumni Home Page
OntoRelSpecifications2	Standard course:	• The Alum clicks on e-mail an alumni link. • The Departmental Server returns a form. • The Alum fills in the form and clicks submit. • The Departmental Server checks to see if any required fields are empty. • If any required fields are empty the Departmental Server returns a message and the form. • If none of the required fields are empty the Departmental Server queries the Alumni Database for the requested Alum's entry. • The Departmental Server returns the non-private information on the requested Alum and a message stating if the requested Alum will accept e-mails. • If the requested Alum is not in the Alumni Database, the Departmental Server returns a message and the Alum is returned to the CIS Home Page. • If the requested Alum will accept e-mails, the Alum can select E-mail this Alum. • If not the Alum can select Search for another Alum or return to CIS Alumni Home Page. • If the Alum chooses to Search for another Alum go to step 2. • If the Alum selects return to CIS Alumni Home Page the Departmental Server returns the Alum to the CIS Alumni Home Page. • The Departmental Server presents the Alum with a form to fill out and a

		place for the message. • The Alum selects send. • The Department Server will forward the e-mail with all necessary information to the requested Alum. • The Departmental Server returns a message and returns the Alum to the CIS Alumni Home Page
	Proxy course:	• None
OntoRelSpecifications3	Exceptions:	• If the connection is terminated before the information is returned, the Departmental Server is returned to the wait state. • If the connection is terminated after the information is returned, the Departmental Server is returned to the wait state
OntoRelSpecifications4	Includes:	• None
	Primacy:	•Medium
	Rate of Use:	•None
OntoRelSpecifications5	Exceptional Requirements:	• None
	Remarks and concern:	• None

5. RESULTS

With the help of case studies, the performance of *OntoReliability protocol* has been examined on the basis of influence of description, preconditions, post conditions, normal flow, alternative flows, exceptions, inclusions, priority, frequency of uses, special requirements and notes and issues with the scale from 1 to 11.

It has been observed that, in AFRS the placing reservation module contains only description, preconditions therefore ORS1 is represented with 1 and 3 values. Similarly ORS2 has both normal flow and alternative flow consequently represented with 4 and 5. Next ORS3 contains all possible exceptions thus signified with 6. Also, ORS4 covers priority and frequency only hence indicated with 8 and 9. Lastly, ORS5 contains both the special requirements and notes and issues thus allotted with 10 and 11 values. Thus, all other modules such as for change/cancel reservation and for Update/Add/Delete Flight information/User reservations has been represented in Figure 3.In addition, Figure 4 represents all modules of WAADS such as to Access Alumni Home Page, of survey, to create new entry, to update an entry, Search for an Alumni/E-mail an Alumni entry.

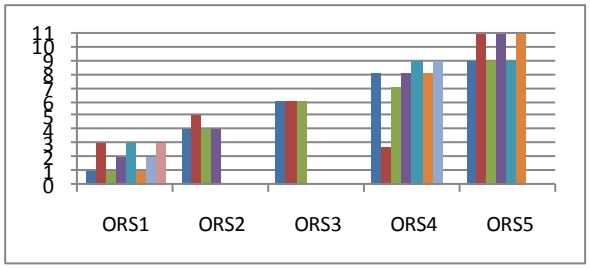

Figure 3: AFRS

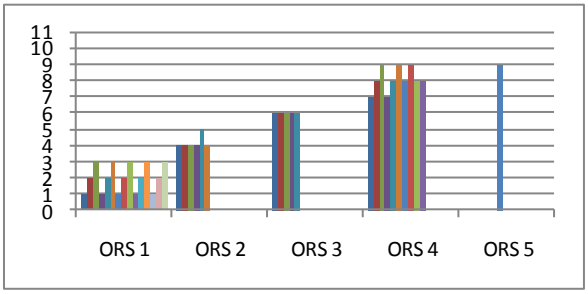

Figure 4: WAADS

6. CONCLUSION

The following statements recapitulate the use of *OntoRel* specifications developed using *OntoReliabilty* protocol.

- *OntoRel* specifications allow demonstrating the behavior of a software system before it is actually implemented. This has three positive consequences for software development.
  - ✓ Executable components are much earlier available than in the traditional life-cycle. Therefore validation errors can be corrected immediately without incurring costly redevelopment.
  - ✓ Requirements that are initially unclear can be clarified and completed by hands-on experience with the executable specifications.
  - ✓ Execution of the specification supplements inspection and reasoning as means for validation. This is especially important for the validation of non-functional behavior.
- *OntoRel* specifications are constructive, these explicitly do not only demand the existence of a solution, conversely actually construct it.
- *OntoRel* specifications do not necessarily constrain the choice of possible implementations because only minimal design and implementation decisions are necessary to get executability. In addition, these decisions are revisable.

7. REFERENCES

[1] J.D. Musa, *Software Reliability Engineering: More Reliable Software Faster and Cheaper (2nd Edition)*, AuthorHouse, 2004.

[2] H. Pham, *Software Reliability*, Springer, Singapore, 2000.

[3] T. Margaria and B. Steffen, "Service Engineering:Linking Business and IT," *IEEE Computer*, October 2006,pp. 45-55.

[4] J. M. Wing, A Specifier's Introduction to Formal Methods, IEEE Computer, Vol. 7, No. 5, pp. 8-24, September 2000

[5] N. Gehani, A. D. McGettrick (Eds.), Software Specification Techniques,Addison-Wesley Publishing Company2000

[6] R. Milner, M. Tofte, R. Harper, The Definition of Standard ML, MIT Press, 2000

[7] W. W. Agresti (Ed.), New Paradigms in Software Development, IEEE Computer Society Press, 2006

[8] M. Chen, M.R. Lyu, and E. Wong, "Effect of Code Coverage on Software Reliability Measurement," *IEEE Transactions on Reliability*, vol. 50, no. 2, June 2001,pp.165-170.

[9] J.D. Musa, "Operational Profiles in Software Reliability Engineering," *IEEE Software*, Volume 10, Issue 2, March 1993, pp. 14-32.

[10] Cortellessa, V., Singh, H., Cukic, B. Eerly reliability assessment of UML based software models. In: Third International Workshop on Software and Performance, Rome, 2007.

[11] Rodrigues, G. N., Rosenblum, D. S., Uchitel. S. Using scenarios to predict the reliability of concurrent component-based software systems. In: 8th International Conference on Fundamental Approaches to Software Engineering, FASE 2005. Springer Lecture Notes in Computer Science, Edinburgh, 2005.

[12] S. Yacoub, B. Cukic, and H. Ammar, "Scenario-based reliability analysis of component-based software," proceeding of the 10th International Symposium on Software Reliability Engineering (ISSRE'99), 1999.

[13] Reussner, R. H., Schmidt, H. W., Poernomo, I. H. Reliability prediction for component-based software architectures. Journal of Systems and Software, 66(3), 241-252, 2003

[14] Grassi, V. Architecture-based dependability prediction for service-oriented computing. In: Proceedings of the Twin Workshops on Architecting Dependable Systems, International Conference on Software Engineering (ICSE 2004), Springer, Edingburgh, 2004.

[15] A. Immonen and E. Niemelä, "Survey of reliability and availability prediction methods from the viewpoint of software architecture," Software and Systems Modeling. 7(1), 49-65, 2008

[16] A. Immonen, "A method for predicting reliability and availability at the architectural level.," in Research Issues in Software Product-Lines - Engineering and Managemen, T. Kakola and J. C. Duenas, Eds. Berlin Heidelberg: Springer Verlag, 2006, pp. 373-422.

[17] E. Niemelä and A. Immonen, "Capturing quality requirements of product family architecture," Information and Software Technology. 49(11-12),1107-1120, 2007.

[18] Shilpa Sharma, Maya Ingle, "An Ontology Aided Requirement Engineering Framework", International Journal of Advanced Research in Computer Science, Volume 2, No. 1, Jan-Feb 2011