INTELLIGENT KNOWLEDGE DATABASE (IKD) TOOL FOR FORMAL METHODS

Manju Nanda[1] J.Jayanthi[2] and Madhan.V[3]

[1]National Aerospace Laboratories, Bangalore, India
manjun@nal.res.in
[2]National Aerospace Laboratories, Bangalore, India
jayanthi@nal.res.in
[3]National Aerospace Laboratories, Bangalore, India
vmadhan86@gmail.com

ABSTRACT

*This paper discusses the I**ntelligent Knowledge Database (IKD) tool** generated for formal methods. The knowledge database provides the information regarding the existing formal methods in the area of academia, industry and R&D sectors. The tool provides complete information about the formal methods adopted in the conventional or model-based approach, in the various phases of the software development life cycle process, list of tools using formal techniques with their version and published literature supporting formal methods. This knowledge-database serves as a live encyclopedia which will enable the engineers and researchers interested in the field of formal methods. The database is intelligent because it provides the user with the flexibility of searching the formal method related information using keyword similar to the search engine. This is a unique tool for formal methods encompassing most of the published literature with intelligent search options.*

KEYWORDS

Formal methods, Information retrieval, Verification and Validation, SDLC Phases, Encyclopedia, Database, Live database tool.

1. INTRODUCTION

In the era of information explosion, information retrieval [1, 5] is the key point in the process of geo-information sharing and integration, and always become a bottleneck. Information retrieval [4] systems handle large amounts of data. Open sources [2] (eg.swish++) also plays a major role in the process of information retrieval. This knowledge-database serves as a live encyclopedia will aid the engineers and researchers interested in the field of formal methods. The database is intelligent because it provides the user with the flexibility of searching the formal method using keyword similar to the search engine [3]. This is a unique tool for formal methods encompassing most of the published literature with intelligent search options. This intelligent search options makes user's search process much simpler and also a user friendly. The main purpose of developing this tool is to provide the required information in a single place to enable the analysis for applying formal methods in a safety critical project/application.

DOI : 10.5121/ijsea.2012.3609

"**Intelligent knowledge Database (IKD)"** is a knowledge based tool which will be helpful to the researchers and engineers who are all currently working or interested to work in the field of formal methods. This tool will act as a unique knowledge based reference tool for all researchers in the future. This tool contains a huge collection of previous references in the form of journal paper, seminar paper, books, and power point presentations images etc.

There are many tools that support formal methods has been information on the best-known and widely used formal tools, with emphasis on tools aimed at industrial practitioners without extensive formal methods expertise. The lack of availability of proper knowledge base on existing formal methods and their application has served as a motivation for generating this database. This database gives an opportunity to all to know and understand the available formal methodologies and their applications along with respective tool support. The database provides comprehensive information on formal methods, tools and their applicability.

The literature survey has led to analysis of various published documents that describe about the applicability (safety critical, commercial and academia) of formal methods. The survey of formal methodologies and their applicability at different phases of design life cycle has been developed into a database. In the existing scenario, this is first software of its kind as there exist no package which provides the encyclopedia of the formal methods being researched, used and implemented across the world.
In this paper, Section 1 gives an introduction about the tool, Section 2 deals with the overview of the IKD, and Section 3 gives the implementation part and section 4 describes the validation of the tool land section 5 deals with the conclusion and future work.

2. OVERVIEW OF THE IKD

The IKD tool consists of two components: Database Architecture which act as a background and the GUI Architecture which act as a front end.

2.1 Database Architecture

The database is developed using the MS-Access 2007. The analysis of the published literatures in the form of journal paper, seminar paper, books, and power point presentations led to an intelligent database tool that serves as a search engine [3]. The database architecture is designed such that the search operation and information retrieval is commercial and fast. This contains large amount of records. Record contains large number of rows and columns. Each record contains various fields and its own subfields. Figure.1 shows the snapshot of the architectural view of the Intelligent Knowledge Database.

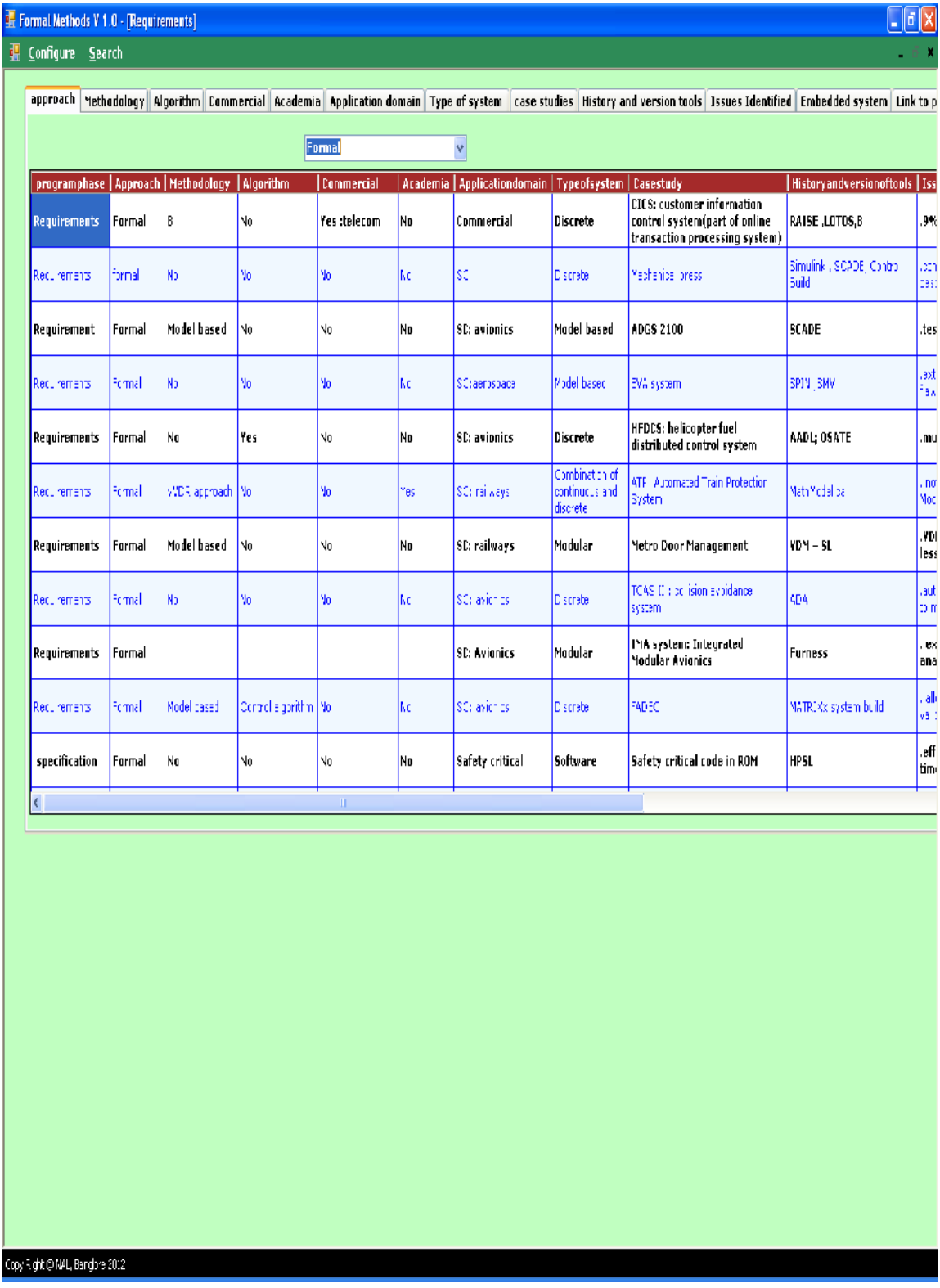

Figure 1: Typical architectural view of the Intelligent Knowledge Database

2.1.1 Record set architecture

IKD contains a large number of records which includes a number of **Tuples (rows)** and **Attributes (columns).** At the end of each row of the record corresponds to a link to the published literature. Column contains fields like approach, methodology, application domain, type of system, case study, history and version of the tools (if applicable), issues identified (if mentioned), embedded system and the link to the published literature.

2.1.1.1 Tuple Structure

Each row contains the software engineering phases like

- Requirements.
- Design.
- Implementation,
- Verification and validation

a. Requirements (Specification) Phase

It defines needed information, function, behavior, performance and interfaces and also defines project goals into defined functions and operation of the intended application. This phase provides all the relevant information which comes under requirements phase. When user selects requirement phase search the search engine [3] gives all the references as the published literatures in the form of journal paper, seminar paper, books, and power point presentations and so on.

b. Design Phase

It describes desired features and operations in detail, including screen layouts, business rules, process, diagrams, pseudo code and other documentation. Data structures, software architecture, interface representations, algorithmic details. This phase provides all the relevant field information like Approach, Methodology, Algorithm, Commercial, Application domain, Type of system, issues identified, history and version of tools, embedded system and so on. Which are all comes under design phase. When user selects design phase search the tool gives all the references as the published literatures in the form of journal paper, seminar paper, books, and power point presentations and so on.

c. Implementation phase

It includes implementation preparation, implementation of the system into product environment and resolution of problems in the previous phases. This phase provides all the relevant field information like Approach, Methodology, Algorithm, Commercial, Application domain, Type of system, issues identified, history and version of tools, embedded system and so on. Which are all comes under implementation phase. When user selects implementation phase search the search engine [3] gives all the references as the published literatures in the form of journal paper, seminar paper, books, and power point presentations and so on.

d. Verification and validation phase

Verification and validation [16] are independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it fulfills its intended purpose. This phase search retrieves information about all type of verification and validation tools, methods.

2.1.1.2 Attribute Structure

Each Attribute has its own subfields. The attributes are approach field, methodology field, algorithm field, application domain field, model based design, History and version field, issues identified field, embedded system field and link to papers field.

Approach field denotes what sort of approach that particular existing tools have been used which can be separated into subfields like formal (B, Z, TRIO, TCOZ), informal and functional.
Methodology field specifies the type of algorithm which has been used in the existing tools which can be divided into Model based, LPPE technique, Reverse Engineering, Distributed RT Model, vVDR approach, MBSE and so on.

Algorithm Field contains the all existing used and supporting algorithms as subfields like scheduling algorithm, Distance Penalty algorithm, URNG Marsalis algorithm and Control algorithm and so on. The most commonly used algorithm is the control algorithm in safety critical applications.

The application domain field describes the boundary of the tool. It may be safety critical, commercial or academia. **Safety critical domain [11]** includes: **Aerospace, avionics, nuclear, medicine, railways, military, automobile, space**. Commercial domain includes: **Telecommunication, communication, networking, and banking**. Academia involves the tools that are being used for research purposes. Mainly involves the support for new add on and innovations that help work in other domains. The case study involves the system under consideration in each of the published literature. The system may be continuous, discrete, combination of both, modular or concurrent.

The model based design [10, 12] field describes the family unit of the tool whether it is a model based tool or not. The different kind of the embedded systems is: **Dependable, Distributed, Real time, Reactive, Safety critical, Mission critical, Modular, Interactive, Data intensive, Control intensive, Reconfigurable, Embedded, Infinite state, Integrated, Parallel, Intelligent, Scalar** etc.

Dependable specifies the parameters involved in the system are dependable or interdependent on some external or internal links. **Distributed** denotes the collection of autonomous computers, connected through a network and distribution middleware, which enables computers to coordinate their activities and to share the resources of the system, so that users perceive the system as a single, integrated computing facility. **Real time** system runs at real time. **Reactive** responds to external event. **Critical failure** of the system may lead to loss which can be minor, major, hazardous or sometimes catastrophic.

Modular subdivides a system into minor parts called modules that can be independently created and then used in different systems to drive multiple functionalities.**Data intensive** Programming abstractions including models, languages and algorithms which allow a natural expression of parallel processing of data. Control intensive resources limitations in terms of memory, bandwidth and energy combined with the existence of dependability and real-time concerns are obviously issues to take into consideration. **Reconfigurable** supports super instruction fetch. Infinite state characterized by unbound data structures and control structures. **Time transition** the qualitative fairness requirements of traditional transition system are replaced (and superseded) by quantitative lower-bound and upper bound timing constraints on transitions. **Intelligent** learns how to act in order to reach its objectives. **Informative** is any combination of information technology and people's activities that support operations, management and decision making. The last field of the tuple contains a link to extract the particular relevant publication. Using these links the users can gets complete information about the formal method.

2.2. GUI Architecture

The GUI architecture of the IKD tool is shown in the Figure 2 as a flowchart which specifies the overall functionality of the tool. The above mentioned GUI was developed using the language **c#** visual studio 2010.

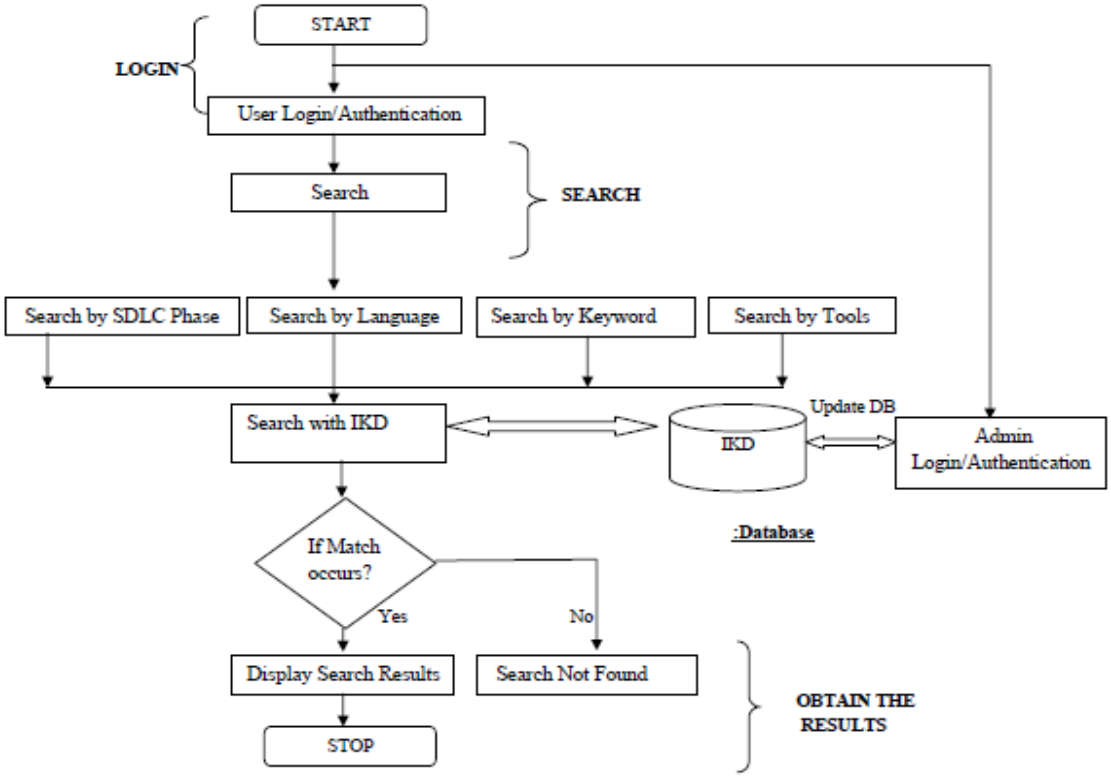

Figure 2. Architecture of the IKD tool

Figure 2 shows the overall functionalities of the tool.

This tool contains four major parts. They are **login**, **search**, **querying the database**, **obtain the results**.

Individual personal username and password are given to all personal users by administrator. Users can login into their account using this login details. After a successful login users are allowed to search and querying the database. Based on the user's inputs the tool gives the results. Updating of the database carried out by the administrator. Users are unacceptable to update the database.

2.3Uniqueness of the tool

The encyclopedia of the formal methods being researched,used and implemented. This IKD tool is portable, simple and user friendly. This tool is live and the database keeps getting updated based on the published public literature available around the world .The tool provides '**Ease of search'** approach where the underlying software simplifies the way of user's searching. The search process is simple and it divides the search process into four categories:

- Search by Phases.
- Search by Language used.
- Search by keyword.
- Search by Tool.

In the search by phase category the search items can be split into software engineering SDLC phases like **Requirement, Design, Implementation** and **verification and validation.** So users may search their items according to phase which they would like to do.

In the search by Language search items are separated into various languages **(e.g. Z method, B method, VDM, AADL, Lustre, SCADE[14], UML, Embedded C, C++,VHDL, SysML[15], PVS, OMEGA, XML, RSML, EJB, Python etc)**used in the model based design and safety critical applications. This type of search will lend a hand to the software developers to develop their software. Also this type of search provides a comparative analysis about the various programming languages.

In the search by keyword category users are requested to give their own keyword which is relevant to formal methods, after getting the keyword the encyclopedia will provide the information which are relevant to the given users keyword. This type of searching guides the new users who are all not aware much about formal methods, model based design and safety critical applications .This method increases the interest level of searching as compared to other searching methods..

Search by tool is the last type of search category which divides the search items into variety of tools e.g. matlab/simulink [9], SCADE [14], UPPAAL, Statemate, Rhapsody, Sysweaver, Mathmodelica,

Openmodelica, Dymola, MapleSim, ASCET, metaEdit, oAW, AUTOFOCUS, Metamill, SynDex, Matrixx etc.

3. TOOL IMPLEMENTATION

The tool was implemented by using the language C# and visual studio 2010 environment. Study of the literature available about the formal methods published [17] was studied in order to develop the information in a way which can be easy to extract and search. This led to the intelligent database that has the features of the search engine. The requirements, design, implementation and the testing of the software to develop the tool followed a simple 'watercycle model' to ensure that the requirements of the tool are captured and implemented correctly. After the implementation the tool was tested and validated. The overall size of the tool is around 200MB. This tool was implemented as a simple and portable one.

Although this tool guides the user by providing relevant feedbacks to the users in case the search item is not found in the database.

4. VALIDATION OF THE DATABASE

The correct implementation of the IKD database was validated by an independent reviewer as per the requirements document. The various test carried out were for the functionality as search by phase, search by language used, search by Manual keyword, search by tools etc. Some of the screen shots of the validation results as given below. **Figure.3** shows the Search by phase validation, **Figure.4** shows the validation result of the Search by Languages used. Similarly **Figure.5 and Figure.6** shows the validation results of the Search by Manual keyword and Search by Tools used respectively.

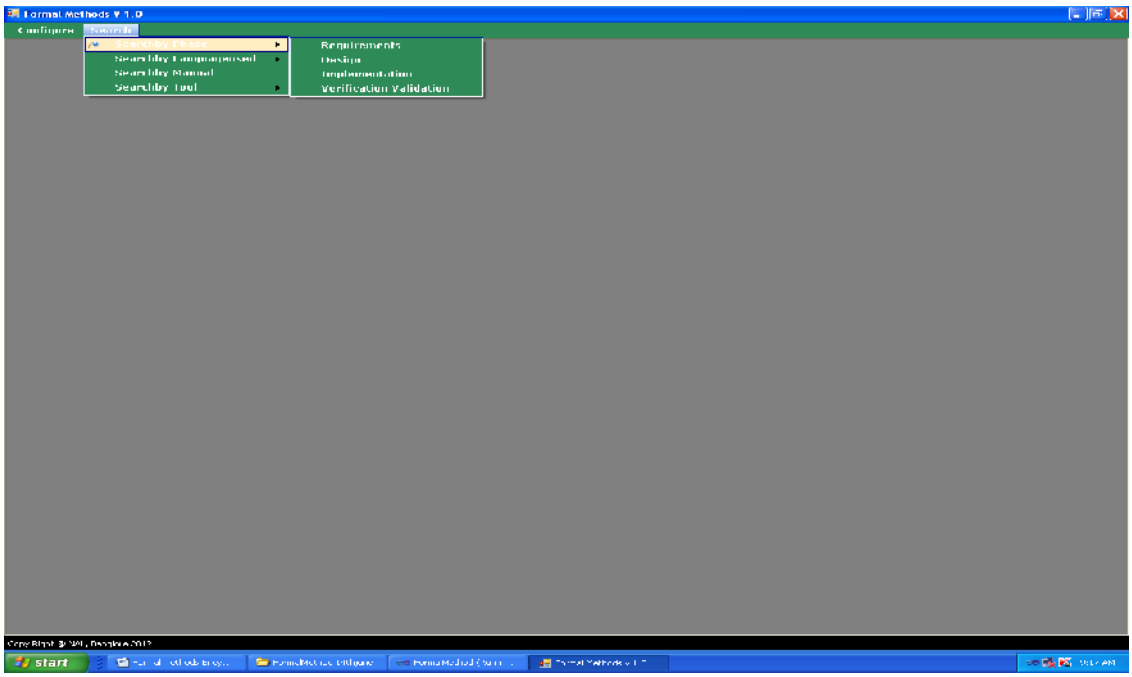

Figure.3 Search by Phase

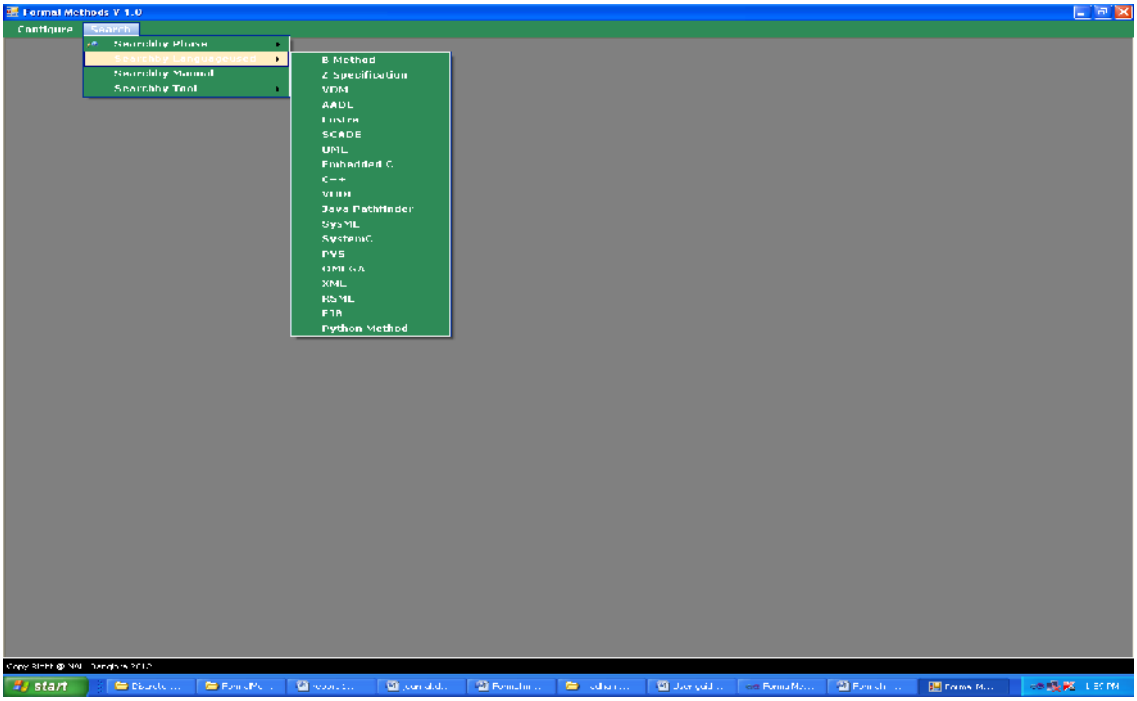

Figure.4 Search by Language

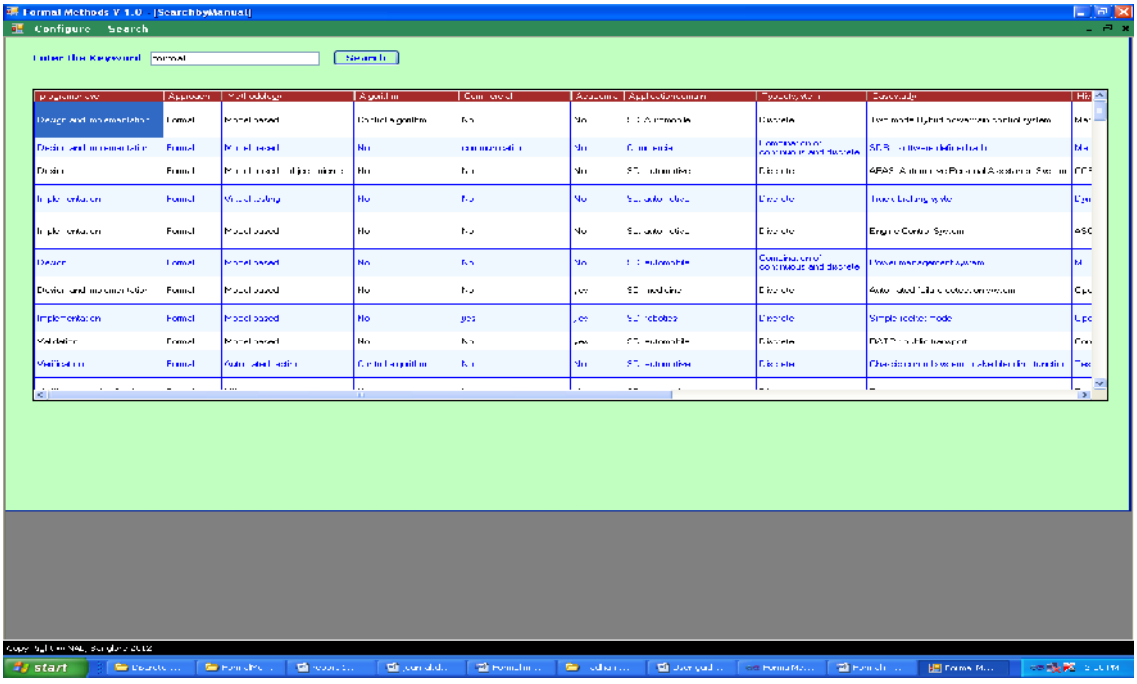

Figure.5 Search by keyword

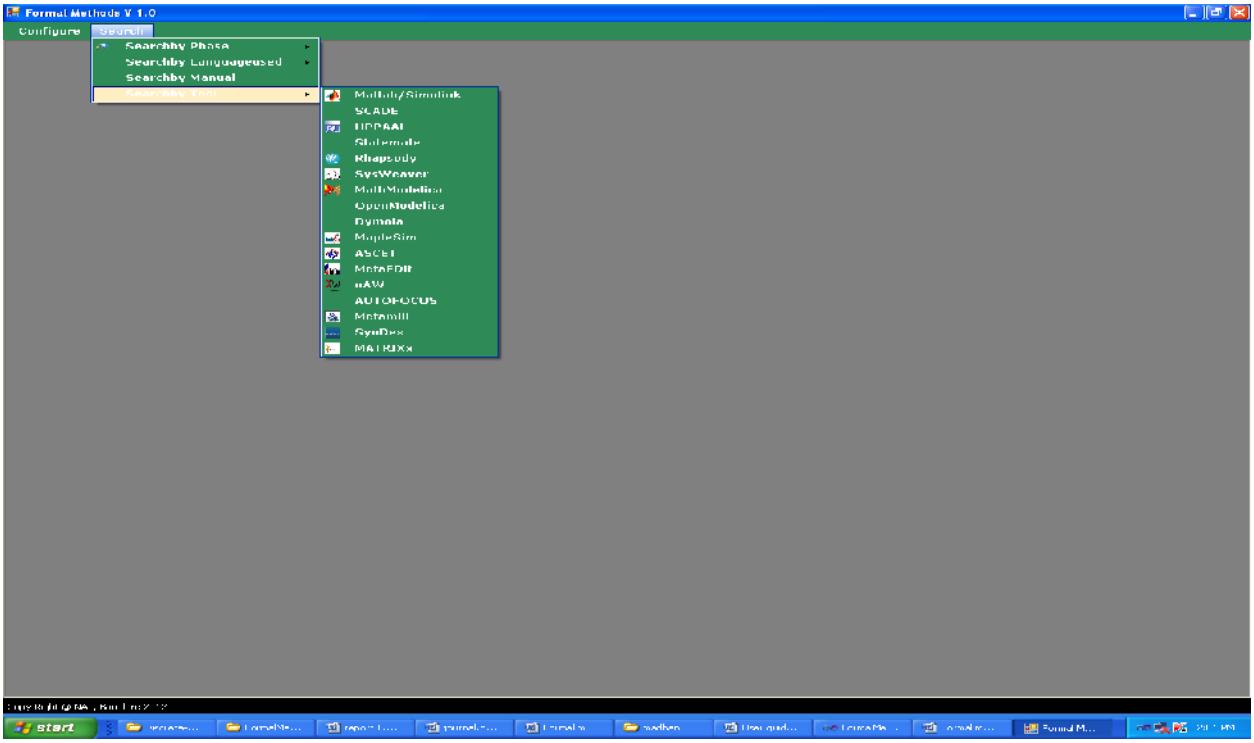

Figure.6 Search by Tool

5. CONCLUSION AND FUTURE WORK

Formal method encyclopedia is a database which containing detailed knowledge and information about a variety of fields or subfields in formal methods. The generated database defines the availability in the present scenario along with the comparative analysis of existing tools.

The future enhancement of the database shall be in the below directions,

- ✓ To add one more search category as search by author name.
- ✓ Updating the database by adding future information.
- ✓ Enhancing the interaction with tool.
- ✓

ACKNOWLEDGEMENTS

This work is a part of the sponsored project from AR&DB. .

REFERENCES

[1] JAMES ALLAN, "Information Retrieval Overview"
[2] ANDY MACFARLANE, "Overview of Open Source and Information Retrieval"
[3] Monika Henzinger , "Tutorial Web Information Retrieval"
[4] Dragos and Anton Manolescu, "Feature Extraction—A Pattern for Information Retrieval"
[5] C.J. "Keith" van Rijsbergen, "Introduction to Information Retrieval"
[6] Morten Hertzum , "A Comparison of Three Data Models for TextStorage and Retrieval Systems"
[7] "Comparison of software metrics tools" by Rüdiger Lincke, Jonas Lundberg and Welf Löwe. Software Technology Group, School of Mathematics and Systems Engineering, Växjö University, Sweden.
[8] "Formal Methods and their role in certification of safety critical systems" by John Rushby.
[9] CONTROL ALGORITHM MODELING GUIDELINES USING MATLAB®, Simulink®, and Stateflow® Version 2.1 Math Works Automotive Advisory Board (MAAB) July 27th, 2007.
[10] "Experiences with formal engineering: Model based specification, implementation and testing of a software bus at Neopost" by M. Sijtema, M.I.A. Stoelinga, A. Belinfante and L.Marinelli: University of Twente, Netherlands and Neopost, Austin, Texas
[11] "Developing Safety-Critical Systems: The Role of Formal Methods and Tools" by Constance Heitmeyer, Center for High Assurance Computer Systems, Washington
[12] "Model Based Design for DO178B with qualified tools" by Tom Errkinen and Bill Potter; Mathworks Inc.
[13] "Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability and Robustness for maintaining Systems Lifecycle value" by Adam M Ross, Donna H Rhodes and Daniel E Hastings (Massachusetts Institute of technology, Cambridge)
[14] "Examination of SCADE in Systems Engineering" by Lars Sarbaek and Martin Kjeldsen. (Engineering College of Aarhus; Denmark).
[15] "Combining SysML and formal models for safety requirements verification" by Jean-François Pétin, Dominique Evrot, Gérard Morel, Pascal Lamy.
[16] Validation and verification of next generation air transportation systems and technologies" by Ronald L Fulton
[17] "A Report on Formal methods encyclopedia" Manju nanda, jayanthi and chinmayi s.jamadhagni