Alloy is a tool for modelling and analysing systems, it consists of two parts: Alloy Language and the alloy Analyzer, It was designed as an attempt to address certain deficiencies in Z notation. It is a small language that can express the basic structure(in space and time) of a system as well as constraints and operations specifying how the system may change. Alloy Language combines boolean algebra, set theory, quantities and first-order relational logic. Alloy is a lightweight formal language that has a simple notation that is easy to learn and use that includes a ready validation and verification tool. The alloy Specification language is based on first-order relational logic with familiar object oriented notation making it easy to classify objects and associate properties with objects according to the classification. Unlike the object orientation which was alloy's original inspiration, the analysis in alloy analyzer relies on boolean Satisfaction (SAT) technology. This technology helps to translate constraints into booleans which are so easy to solve making alloy tools to be a faster tool as compared to other specification tools. The alloy automatic analyzer enables building of a continuing model with automatic review and checking for errors as you go along developing giving you a sense of confidence experience in modelling. Alloy is designed with automatic analysis in mind; it differs from many specification languages designed for model-checking in that it permits the definition of infinite models. The analyzer performs finite scope checks even on finite models.

There are two approaches to formal specification that are used in industrial software systems and these are; 1. An algebraic approach in which the system is described in terms of operations and their relationships and 2. A model-based approach in which a model of a system is built using mathematical constructs such as sets and sequences and the system operations are defined by how they modify the system state.

Alloy uses relational logic specification and allows automation by formally restricting operations to a limited set of behaviours and data types which is implemented in the alloy automatic analyzer tool. In modelling relationships, this is used to define the relations and sets which define how they affect the state of the system model.

Relations is the fundamental building block of alloy, literally everything in alloy is a set of relations. Taking example above, Users, Account and Resource are useless without the relations between them. The relations are used to map different entities with a communication defined in the given model. These relations are represented as sigs with attributes

Large systems are decomposed into subsystems with well defined interfaces between these subsystems Specification of subsystem interfaces allows independent development of the different subsystems Interfaces may be defined as abstract data types or object classes The algebraic approach to formal specification is particularly well-suited to interface specification

Advantages of using Alloy

- Alloy is very expressive and is often very similar to the codes used in the programming languages

- The alloy tool will find errors in a specification that were initially overlooked which will lead to less maintenance of the system.
- Alloy produces an abstract model of a system which will make it much easier to evolve, or expand on that system in the future: Thus, Alloy can check to ensure that the modifications of a system are compatible with its original specification.
- Alloy's use of ASCII characters makes it compatible with most computer systems.

Disadvantages of Alloy

- The highly technical nature of alloy makes it difficult for users who are not well-educated in mathematical languages
- Analyses in the alloy analyzer need to be limited to a specific scope
- The Analyzer's use of counter-examples may show that the system has flaws but it doesn't prove that the system works properly.

Alternatives to Alloy

Alloy is only one of several approaches to the modelling and analysis of software abstractions. Depending on the ability to capture complex structure in abstract way, these four can be alternatives to alloy B, OCL, VDM and Z These four tools have common features such as offering a notation that can capture software abstractions more directly than a programming language can.They all view state in terms of classical mathematical structures despite the difference in syntax and semantics.They all have support for lightweight tools for evaluating constraints against concrete cases.

At the same time there are crucial differences with the Alloy. In B , VDM, Z an OCL, a notion of state machine is hardwired, in contrast, which is designed to support a variety of idioms, each as easy or difficult to express as the others, B, VDM and Z were designed more with proof in mind than lightweight analysis and so unlike Alloy that is supported by theorem provers.

References

1. Lamsweerde, A. V. (2000, May). Formal specification: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering* (pp. 147-159).
2. Transmuting Base Alloy Specifications into Implementations Shriram Krishnamurthi Brown University Daniel J. Dougherty WPI Kathi Fisler WPI Daniel Yoo WPI
3. TEST GENERATION FROM BOUNDED ALGEBRAIC SPECIFICATIONS USING ALLOY Work supported by FCT under contract PTDC/EIA/103103/2008
4. Software Abstractions Logic, Language, and Analysis Daniel Jackson
5. Alloy: A Language and Tool for Exploring Software Designs BY DANIEL JACKSON