*The **Constraint Solver**, developed at the IBM Haifa Research Lab, is the core technology behind IBM's hardware design verification tools. IBM development labs around the world employ thousands of seats of these tools to verify new hardware designs.*

# IBM Research Constraint Solver

## Constraint Solver Features

The **Constraint Solver** is a robust general-purpose tool that has been used for more than a decade in modeling and solving many complex problems. It uses both deterministic and stochastic searches:

- **GEC**, the **deterministic engine** performs systematic search with backtracking through arc consistent states.
- **Stocs**, the **stochastic engine** relies on learning the high level parameters of the problem characteristics and uses a non-local escape algorithm.

The **Constraint Solver** is adept at handling:

- Up to hundreds of thousands of variables and constraints
- Variables with very large domains
- Complex relations between large numbers of variables
- Conditioned constraints contingent on the value of the variables
- Parameterized constraint problems

## Constraint Solver in Action

Over the last decade, the **Constraint Solver** has been used to verify processors for:

- IBM iSeries, pSeries, and zSeries servers
- IBM Cell processor
- Sony, and Toshiba, and Tensilica

IBM's hardware design verification tools have reduced design and verification costs by more than $100M.

IBM has used the **Constraint Solver** to model and solve complex problems involving vehicle configuration, workforce management, and floor planning.

## Constraint Solver Examples

### Rules Configuration for Automotive Industry

Modern vehicles are highly configurable and must be shipped according to strict customer specifications. In addition, many types of rules govern the validity of a configuration. These include manufacturing, engineering, marketing, legal, and other types of rules. Altogether, the configuration of a vehicle must conform to tens of thousands different rules.

This complexity is coupled with the tight business pressure prevalent in the automotive industry. Consequently, state-of-the-art technology needs to be applied when seeking a valid configuration that conforms to user requirements. Constraint satisfaction technology from IBM Research does exactly that.

In a recent win, IBM Global Business Services (GBS) in Germany worked closely with IBM Research in Haifa to propose a total solution for the truck configuration problem to a large automotive company.

The proposal, based on the IBM Research **Constraint Solver**, enabled validation and default completion of rules, along with powerful capabilities used to extend partial configurations by solving fully-fledged constraint satisfaction problems (CSPs).

In the proposal stage, IBM triumphed over strong competition by demonstrating how constraint satisfaction technology can provide the highest value to the customer. The resulting deal proves the aptness of the approach to modern industry and opens a new window of opportunity for collaboration with other major players in the automotive industry.

## Workforce Management

In a typical workforce management problem, thousands of highly-skilled individuals must be matched and assigned to thousands of jobs. Both the individuals and the jobs impose many constraints on the required and available job role, set of skills, pay level, geographic location, availability, formation of teams, and many more. In addition to solving the resulting constraint problem the solver may also relate to constraints imposed by the training required for an individual to fit a job.

Satisfying all constraints increases the overall number and quality of assignments, thus significantly reducing corporate costs and increasing employee job satisfaction.

## Floor Planning

When a number of hardware units must be positioned on a single chip, constraints apply to the type of each unit, its position, the wiring between units, and the placement of auxiliary units necessary to improve the chip's functionality.

Satisfying the floor planning constraints optimally leads to faster and more powerful chips for use in high-end electronic devices.

## Floating Point Verification

Floating point verification problems typically include arithmetical operations between two operands and constraints on the range, mask, and bit counts.

For example, assuming that `a`, `b`, and `c` are 128-bit integers, a system may include these constraints:

- `a` is in the range `[A,B]` for constants `A` and `B`
- `b` is constrained by the mask `101x101 … 101x101` where `x` represents an unconstrained bit
- `c` has one bit between the constants `Nmin` and `Nmax`
- `c = a` multiplied by `b`

## Processor and System Verification

Typically, a processor specification that includes hundreds of machine instructions is modeled as constraints, thereby allowing the **Constraint Solver** to generate valid tests.

Constraints range from simple bitwise relations between 64-bit integers to complex semantic relations that enforce address translation. In a constraints system, the situation becomes even more complex as dozens of components impose their constraints simultaneously.

To find out more about the **IBM Research Constraint Solver**, contact **Yehuda Naveh** naveh@il.ibm.com
http://www.haifa.il.ibm.com/dept/svt/simulation_vsml_octopus.html