



The HiVy Toolset

Paula J. Pingree

**Jet Propulsion Laboratory (JPL)
4800 Oak Grove Drive
Pasadena, CA 91109**

Paula.J.Pingree@jpl.nasa.gov

Erich Mikk

Erlangen, Germany

Erich.Mikk@epost.de

1.0 Introduction

The **HiVy** toolset provides model checking for statecharts ([SFUG]). This is achieved by translating statechart specifications into the input language of the Spin model checker ([Hol97]). The **HiVy** toolbox transforms output of the commercial tool Stateflow® provided by The Mathworks. **HiVy** can also be used independently from Stateflow. An abstract syntax of hierarchical sequential automata (HSA) is provided as an intermediate format for the toolset [Mik02]. The **HiVy** toolset programs include *SfParse*, *sf2hsa*, *hsa2pr* and the HSA merge facility.

2.0 Constructing a Statechart

First a statechart model of the system to be verified must be constructed. Access to the Stateflow application and general familiarity with the tool is needed. **HiVy** supports a sub-language of

statecharts. This section defines the syntactic and semantic constraints that statecharts must satisfy.

Syntactic restrictions. In order to use the **HiVy** toolbox the statechart model must be designed in a sub-set of the statechart language. This sub-set does not support the following:

- Generation of events
- Inner transitions with the same source and destination
- Backtracking, i.e., every junction must have at least one enabled emerging transition to comply syntactically
- Transition actions on transition segments that end on a junction
- History junctions

Semantic restrictions - Scoping rules. Stateflow scoping rules dictate where the types of non-graphical objects can exist in the hierarchy. Stateflow allows for local state and event names, however **HiVy** does not support this feature. Instead, all state and event names are assumed to be global. In order to comply with this assumption, all state and event names must be defined in the top-level statechart of the model.

Support for embedded statecharts. Stateflow allows the use of embedded statecharts called subcharts. Subcharts enable you to reduce a complex chart to a set of simpler, hierarchically organized diagrams. In order to use this feature safely with **HiVy**, the name of the reference must coincide with the top-most state name of the referenced sub-chart.

Adapting the statechart model for verification. Leveraging on the Spin verification system, **HiVy** supports verification of closed systems only, i.e. the specification to be verified must contain a model of the environment as well.

3.0 Preparing Input for the Compiler

This section describes how to extract statecharts from Stateflow, parse extracted models and merge subcharted statecharts with their parent charts.

Model Files. Statechart design representations are captured in Stateflow model files.

Parsing. Two programs of the **HiVy** toolbox: **SfParse** and **sf2hsa** are used to prepare the model file for translation. If parsing is successful, a file is produced that contains an ASCII representation of the abstract syntax tree in HSA-format ([Mik02]).

Alternate approach to creating HSA. It is not required to have an entirely graphical statechart representation of the system for verification. It may in some cases be simpler to specify a component of the system in tabular notation in which the states, transitions, hierarchy, default transitions, etc. are captured. In this manner the tool extension 'xl2hsa' can be used to convert Excel specifications into HSA.

4.0 Translation

Once the components of the system are parsed in HSA **HiVy** generates Promela input for the Spin model checker.

4.1 Merging Statecharts

If the model consists of several files, then they may be merged into one HSA file before translating into Promela for Spin using the **HiVy** program **hsacomplete**.

On name conflicts and how they are avoided. As discussed in Section 2.0, Stateflow allows for local names (with a notion of the name scope). To preserve scope when merging a subcharted statechart with its parent, all state names are extended by the name of the root state of the subchart. This resolves potential name clashes in the merged statechart.

It is important to know these naming conventions because during verification, the user is provided with propositions that refer to renamed states. These propositions are the means for formalizing LTL properties about the statechart model for Spin.

4.2 The HSA to Promela Compiler: **hsa2pr**

The program **hsa2pr** is used to generate Promela code from the .hsa file. The following files are generated by **hsa2pr**:

- **stmodel.pr**: the Promela model of the original statechart.
- **propositions**: contains names and definitions of propositions. One proposition is generated for each state and each event.

- prop list: contains just the names of propositions (not their definitions). These proposition names are suitable for automatic generation of LTL properties during verification.

The auto-translated file **stmodel.pr** contains an include statement for a file named *never*. This file contains the Spin “never claim” to be verified. The never claim is not generated by hsa2pr and must be created before applying Spin to the generated model.

5.0 Verification

The full capability of the Spin model checker may be used to verify models generated by **HiVy** translation because they yield Promela code. The validity and usability of **HiVy** generated models for Spin model checking has been prototyped on spacecraft Fault Protection designs [PMHSD02].

6.0 System Requirements

To run **HiVy**, the following software is required:

1. **HiVy** Toolset is distributed in source form and should run on any UNIX workstation. Contact Paula.J.Pingree@jpl.nasa.gov.
2. Spin verification system
(<http://netlib.bell-labs.com/netlib/Spin/whatiSpin.html>)
3. Stateflow® is commercial software provided by The Mathworks. (<http://www.mathworks.com/>). Note: Without access to Stateflow, the HSA format may be used to pass specifications to **HiVy** compilers.

References

- [Hol97] G.J. Holzmann. The Model Checker Spin. IEEE Trans. on Software Engineering, 23(5):279-295, May 1997. Special issue on Formal Methods in Software Practice.
- [Mik02] E. Mikk. HSA-Format, *private communication* 2002.
- [PMHSD02] P. Pingree, E. Mikk, G. Holzmann, M. Smith, D. Dams, Validation of Mission Critical Software Design And Implementation Using Model Checking. The 21st Digital Avionics Systems Conference, October 2002
- [SFUG] The Mathworks Stateflow Users Guide, <http://www.mathworks.com>



The HiVy Toolset

Paula J. Pingree

Jet Propulsion Laboratory (JPL)
4800 Oak Grove Drive
Pasadena, CA 91109

Paula.J.Pingree@jpl.nasa.gov

Erich Mikk

Erlangen, Germany

Erich.Mikk@epost.de

1.0 Introduction

The **HiVy** toolset provides model checking for statecharts ([SFUG]). This is achieved by translating statechart specifications into the input language of the Spin model checker ([Hol97]). The **HiVy** toolbox transforms output of the commercial tool Stateflow® provided by The Mathworks. **HiVy** can also be used independently from Stateflow. An abstract syntax of hierarchical sequential automata (HSA) is provided as an intermediate format for the toolset [Mik02]. The **HiVy** toolset programs include *SfParse*, *sf2hsa*, *hsa2pr* and the HSA merge facility.

2.0 Constructing a Statechart

First a statechart model of the system to be verified must be constructed. Access to the Stateflow application and general familiarity with the tool is needed. **HiVy** supports a sub-language of

statecharts. This section defines the syntactic and semantic constraints that statecharts must satisfy.

Syntactic restrictions. In order to use the **HiVy** toolbox the statechart model must be designed in a sub-set of the statechart language. This sub-set does not support the following:

- Generation of events
- Inner transitions with the same source and destination
- Backtracking, i.e., every junction must have at least one enabled emerging transition to comply syntactically
- Transition actions on transition segments that end on a junction
- History junctions

Semantic restrictions - Scoping rules. Stateflow scoping rules dictate where the types of non-graphical objects can exist in the hierarchy. Stateflow allows for local state and event names, however **HiVy** does not support this feature. Instead, all state and event names are assumed to be global. In order to comply with this assumption, all state and event names must be defined in the top-level statechart of the model.

Support for embedded statecharts. Stateflow allows the use of embedded statecharts called subcharts. Subcharts enable you to reduce a complex chart to a set of simpler, hierarchically organized diagrams. In order to use this feature safely with **HiVy**, the name of the reference must coincide with the top-most state name of the referenced sub-chart.

Adapting the statechart model for verification. Leveraging on the Spin verification system, **HiVy** supports verification of closed systems only, i.e. the specification to be verified must contain a model of the environment as well.

3.0 Preparing Input for the Compiler

This section describes how to extract statecharts from Stateflow, parse extracted models and merge subcharted statecharts with their parent charts.

Model Files. Statechart design representations are captured in Stateflow model files.

Parsing. Two programs of the **HiVy** toolbox: **SfParse** and **sf2hsa** are used to prepare the model file for translation. If parsing is successful, a file is produced that contains an ASCII representation of the abstract syntax tree in HSA-format ([Mik02]).

Alternate approach to creating HSA. It is not required to have an entirely graphical statechart representation of the system for verification. It may in some cases be simpler to specify a component of the system in tabular notation in which the states, transitions, hierarchy, default transitions, etc. are captured. In this manner the tool extension ‘xl2hsa’ can be used to convert Excel specifications into HSA.

4.0 Translation

Once the components of the system are parsed in HSA **HiVy** generates Promela input for the Spin model checker.

4.1 Merging Statecharts

If the model consists of several files, then they may be merged into one HSA file before translating into Promela for Spin using the **HiVy** program **hsacomplete**.

On name conflicts and how they are avoided. As discussed in Section 2.0, Stateflow allows for local names (with a notion of the name scope). To preserve scope when merging a subcharted statechart with its parent, all state names are extended by the name of the root state of the subchart. This resolves potential name clashes in the merged statechart.

It is important to know these naming conventions because during verification, the user is provided with propositions that refer to renamed states. These propositions are the means for formalizing LTL properties about the statechart model for Spin.

4.2 The HSA to Promela Compiler: **hsa2pr**

The program **hsa2pr** is used to generate Promela code from the .hsa file. The following files are generated by **hsa2pr**:

- **stmodel.pr**: the Promela model of the original statechart.
- **propositions**: contains names and definitions of propositions. One proposition is generated for each state and each event.

- prop list: contains just the names of propositions (not their definitions). These proposition names are suitable for automatic generation of LTL properties during verification.

The auto-translated file **stmodel.pr** contains an include statement for a file named *never*. This file contains the Spin “never claim” to be verified. The never claim is not generated by hsa2pr and must be created before applying Spin to the generated model.

5.0 Verification

The full capability of the Spin model checker may be used to verify models generated by **HiVy** translation because they yield Promela code. The validity and usability of **HiVy** generated models for Spin model checking has been prototyped on spacecraft Fault Protection designs [PMHSD02].

6.0 System Requirements

To run **HiVy**, the following software is required:

1. **HiVy** Toolset is distributed in source form and should run on any UNIX workstation. Contact Paula.J.Pingree@jpl.nasa.gov.
2. Spin verification system
(<http://netlib.bell-labs.com/netlib/Spin/whatiSpin.html>)
3. Stateflow® is commercial software provided by The Mathworks. (<http://www.mathworks.com/>). Note: Without access to Stateflow, the HSA format may be used to pass specifications to **HiVy** compilers.

References

- [Hol97] G.J. Holzmann. The Model Checker Spin. IEEE Trans. on Software Engineering, 23(5):279-295, May 1997. Special issue on Formal Methods in Software Practice.
- [Mik02] E. Mikk. HSA-Format, *private communication* 2002.
- [PMHSD02] P. Pingree, E. Mikk, G. Holzmann, M. Smith, D. Dams, Validation of Mission Critical Software Design And Implementation Using Model Checking. The 21st Digital Avionics Systems Conference, October 2002
- [SFUG] The Mathworks Stateflow Users Guide, <http://www.mathworks.com>