From Statecharts to Test Case Generation on Web

Alessandro Oliveira Arantes[1,2], Nandamudi L. Vijaykumar[3], Valdivino Alexandre de Santiago Junior[4]

[1]Institute for Advanced Space Studies (IEAv) – Aerospace Technological Center (CTA)
P. O. Box 6044 – 12228-970 – São José dos Campos – SP – Brazil

[2]Student, Graduate Program in Applied Computing (CAP)
[3]Associated Laboratory of Computing and Applied Mathematics (LAC)
[4]Department of Atmospheric and Space Sciences (CEA)
National Institute for Space Research (INPE),
P. O. Box 515 – 12245-970 – São José dos Campos – SP – Brazil

alessandro.arantes@ieav.cta.br,vijay@lac.inpe.br

Abstract. *The development of web-based applications is becoming common since it is a low cost solution which enables teams spread over a country or even over the world to work together in order to implement the product. This work presents a web hosted application, WebPerformCharts, which implements PerformCharts tool to enable software testers to generate test sequences remotely via Internet. The objective of WebPerformCharts is to generate test sequences according to the specification of a reactive system in Statecharts using such resources as an on-line database and web-site user interface.*

1. Introduction

Software development in geographically distributed settings is a trend where many software companies are using computer-supported cooperative tools to overcome the geographical distance and reducing development costs. Within this scenario, web-based applications can aid different teams to work cooperatively in process activities related to the software development life cycle. It is a very important phase especially when dealing with complex software applications such as space applications.

This work presents a web-based application that incorporates two main features: (i) an alternative to FSM (Finite State Machines) by using Statecharts to generate test sequences; (ii) a web-based application that test designers can generate and obtain test sequences remotely via Internet, addressing distributed software development situations.

2. Generating Test Sequences from Statecharts

Statecharts are graphical-oriented and can specify reactive systems and they are formal ([Harel 1987] and [Harel and Pnueli 1987]) enabling to be handled computationally. They extend state-transition diagrams with notions of hierarchy and orthogonality and can be clustered to represent depth. Events change system behavior so that configurations move to other configurations, and they are classified into internal and external [Vijaykumar, 2002].

PerformCharts tool uses Statecharts in test sequence generation, and it can be used for two purposes: (i) obtain performance evaluation by converting the Statecharts model into a Markov chain; (ii) obtain test sequences by converting the model into a FSM. This FSM is the basis to generate test sequences using yet another tool CONDADO [Martins 2000], developed by the State University of Campinas (UNICAMP).

A text-based interface PcML (Performcharts Markup Language) [Amaral 2004], based on XML (EXtensible Markup Language) [W3C 2002], has been developed as a user friendly environment to specify Statecharts diagrams.

3. WebPerformCharts

WebPerformCharts has been proposed to enable different teams working in software through a web-based interface and database application via Internet. It is under development and uses technologies as HTML (HyperText Markup Language), PHP (Hypertext PreProcessor) and MySQL. Therefore, it could be hosted in Apache server and Linux operational system in order to be entirely free of costs with software packages. Also, the clients can use low cost hardware since all processing activities occur in the server, and it (server) must have enough performance in order to respond to several simultaneous accesses.

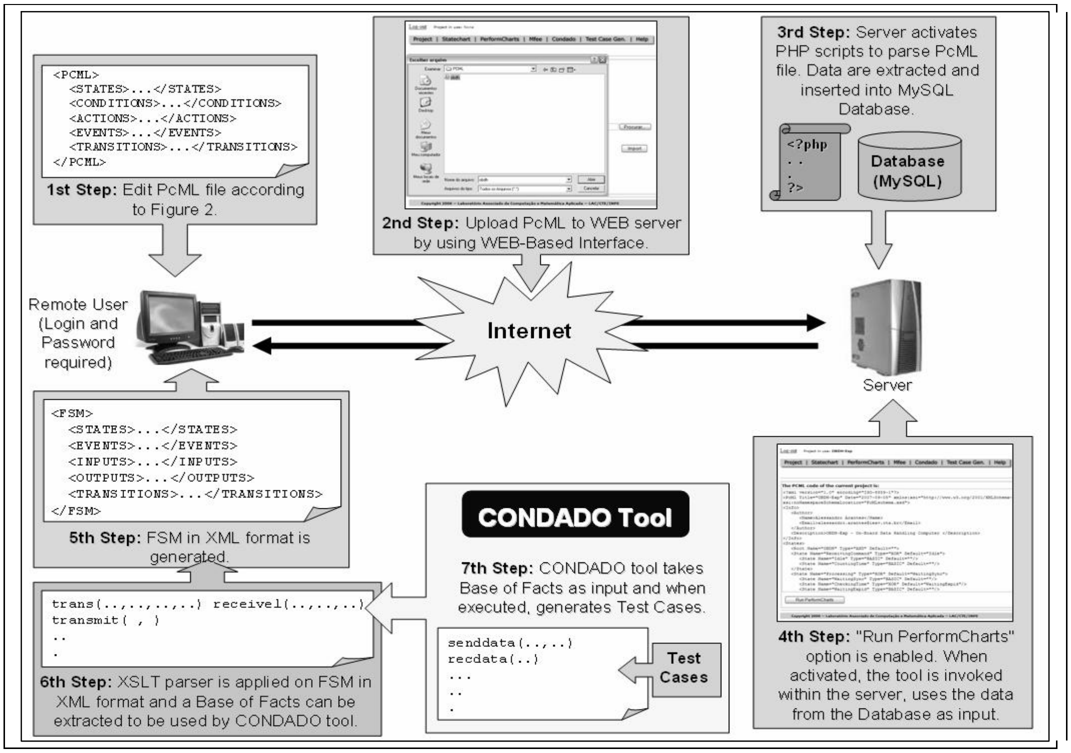
The first phase of WebPerformCharts reads a PcML specification from an editor or uploaded to the server. Scripts (in PHP) parse PcML and extract all relevant data which is stored in a database from which the necessary data structures (to hold the encapsulation, states, events, conditions, parallel components and transitions) are created as well as the calls to appropriate methods to generate the FSM. When performance evaluation is required, a Markov chain is the result. Then, the output of the state-transition diagram is stored in the database and can be extracted in XML format for any other use. The summary of these steps is shown in Figure 1. CONDADO tool accepts an FSM specified as a base of facts. Therefore, the FSM in XML format has to be converted into this base of facts. This conversion is achieved by using a XSLT (Extensible Stylesheet Language Transformation) parser.

In a second phase, other methods to generate test sequences will also be integrated into the application; there are several other suggestions of minimizing the FSM from Graph theory, and they will be considered for implementation enabling the comparison of results from different methods. In addition, work is in progress to generate test sequences directly from Statecharts specification instead of converting into FSM. This means that in Figure 1, the system will jump from 4th step directly to 7th step.

4. Conclusions

Web-based systems have many advantages than a conventional system, since it can be accessed in real time conditions from any place in the world at anytime with a computer or laptop, an internet connection and a web browser. Also, importing and exporting XML data open doors to integrate other test sequence generation methods in future since the XML language is standard and relatively easy to learn.

The main contribution of this work is to enable PerformCharts for supporting the test process in a cooperative environment. As a second contribution, the implementation of other methods to generate test sequences will enable comparison of results from different methods, besides their analysis.



```
<PCML>
  <STATES>...</STATES>
  <CONDITIONS>...</CONDITIONS>
  <ACTIONS>...</ACTIONS>
  <EVENTS>...</EVENTS>
  <TRANSITIONS>...</TRANSITIONS>
</PCML>
```

1st Step: Edit PcML file according to Figure 2.

2nd Step: Upload PcML to WEB server by using WEB-Based Interface.

3rd Step: Server activates PHP scripts to parse PcML file. Data are extracted and inserted into MySQL Database.

```
<?php
. .
.
?>
```

4th Step: "Run PerformCharts" option is enabled. When activated, the tool is invoked within the server, uses the data from the Database as input.

```
<FSM>
  <STATES>...</STATES>
  <EVENTS>...</EVENTS>
  <INPUTS>...</INPUTS>
  <OUTPUTS>...</OUTPUTS>
  <TRANSITIONS>...</TRANSITIONS>
</FSM>
```

5th Step: FSM in XML format is generated.

```
trans(..,..,..,..) receivel(..,..,..)
transmit( , )
..
.
```

6th Step: XSLT parser is applied on FSM in XML format and a Base of Facts can be extracted to be used by CONDADO tool.

7th Step: CONDADO tool takes Base of Facts as input and when executed, generates Test Cases.

```
senddata(..,..)
recdata(..)
...
..
.
```

Figure 1. WebPerformCharts Architecture

References

Amaral, A.S.M.S.; Veloso, R.R.; Vijaykumar, N.L.; Francês, C.R.L.; Oliveira, E. (2004) "On proposing a Markup Language for Statecharts to be used in Performance Evaluation", International Journal of Computational Intelligence, 1(3), pp. 260-265.

Harel, D. (1987) "Statecharts: a visual formalism for complex systems", Science of Computer Programming , Vol.8., pp. 237-274.

Harel, D.; Pnueli, A.; Schmidt, J.; Sherman, R. (1987) "On the formal semantics of Statecharts", IEEE Symposium on Logic in Computer Science, Ithaca, USA.

Harel, D.; Politi, M. (1998) "Modeling Reactive Systems with Statecharts: the Statemate Approach", McGraw-Hill, USA.

Martins, E.; Sabião, S. B.; Ambrósio, A. M. (2000) "Condata: a tool for automating specification-based test case generation for communication systems". 33rd Hawaii International Conference on System Sciences.

Vijaykumar, N.L.; Carvalho, S.V.; Abdurahiman, V. (2002) "On proposing Statecharts to specify Performance Models", International Transactions in Operational Research, 9(3), pp. 321-336.

W3C (2002) Extensible Markup Language (XML) http://www.w3.org/XML/Activity.