CELTIC
Telecommunication Solutions

Gandalf

Number: D6.2

Demonstrate Automatic Diagnosis in Troubleshooting in a Prototype Product

Contractual Date of Delivery: January 2007
Actual Date of Delivery: March 2007
Author(s): See list below
Editor: Lars Moltsen
Participant(s): Wirtek (Moltsen Intelligent Software), France Telecom R&D, Telefonica I+D, University of Limerick
Workpackage: WP6
Nature: *(Public)*
Version: v1.0
Total number of pages: 14

Abstract:

This deliverable provides an exact listing of the functionality of the developed auto-troubleshooting toolset. In addition, an evaluation of the toolset on practical trials carried out in the project is presented. The conclusion of the results (of this deliverable as well as the quantitative results in D5.3a [2]) is that by using Bayesian network technology, we have produced a toolset capable of saving operators a huge amount of OPEX on the manual work in the area of troubleshooting.

Keyword list: Software, Demonstration, Validation, Questionnaire, Automation, Automated Troubleshooting, Automated Diagnosis, Bayesian Network

LIST OF AUTHORS

Lars Moltsen	Wirtek (Moltsen Intelligent Software)
Jordi Triola Bosch	France Telecom R&D
Beatriz Solana	Telefonica I+D
Carlos de Antonio Ramos	University of Limerick

Executive Summary

This deliverable provides an exact listing of the functionality of the developed auto-troubleshooting toolset. In addition, an evaluation of the toolset on practical trials carried out in the project is presented. The conclusion of the results (of this deliverable as well as the quantitative results in D5.3a [2]) is that by using Bayesian network technology, we have produced a toolset capable of saving operators a huge amount of OPEX on the manual work in the area of troubleshooting.

Acronyms

ADFF: Auto-Diagnosis File Format
KPI: Key Performance Indicator
OPEX: Operational Expenditures
XML: Extensible Markup Language

Table of Contents

List of Figures:

List of Tables

1 Introduction

The development of the software demonstration for automated troubleshooting has practically been overlapping a lot with development of the core auto-troubleshooting software described in D5.2 [1] and with results presented in D5.3a [2]. This document focus on the exact functionality of the available demo software as well as performance evaluation of this toolset.

In Section 2 the functionality of the auto-troubleshooting toolset is listed. In Section 3, the results of a feedback questionnaire exercise is presented. In Section 4 we derive some conclusions on the quality and usability of the demo toolset based on the feedback.

2 Functionality

2.1 Toolset Overview

A toolset has been developed consisting of two individual tools:

1. “Knowledge Builder” enables the user to specify a model from expert knowledge and/or from available learning data
2. “TheCure” enables the user to use a model for computing a diagnosis given a set of input data (performance indicator readings).

Figure 1 shows a screen dump of “Knowledge Builder”, where a number of fault causes are shown to the upper left, and a number of symptoms are shown to the lower left.

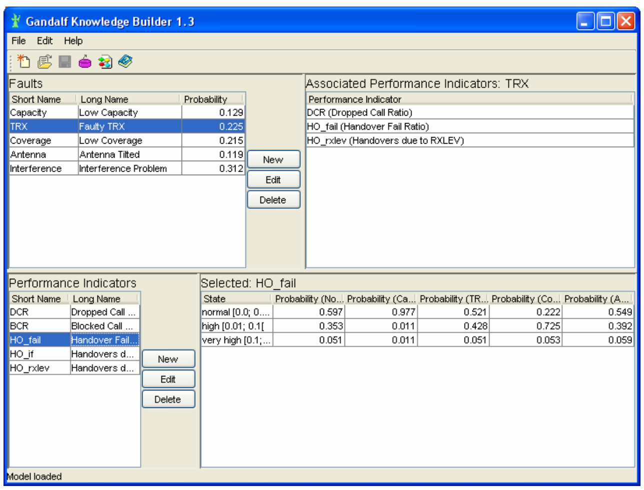

Figure 1. The “Knowledge Builder” tool

Figure 2 shows “TheCure” currently with 5 cells loaded. To the lower right the *evidence* (observed KPI values) is shown for the cell selected in the left panel. Above the evidence is the actual diagnosis – the computed probabilities of fault cases given the evidence.

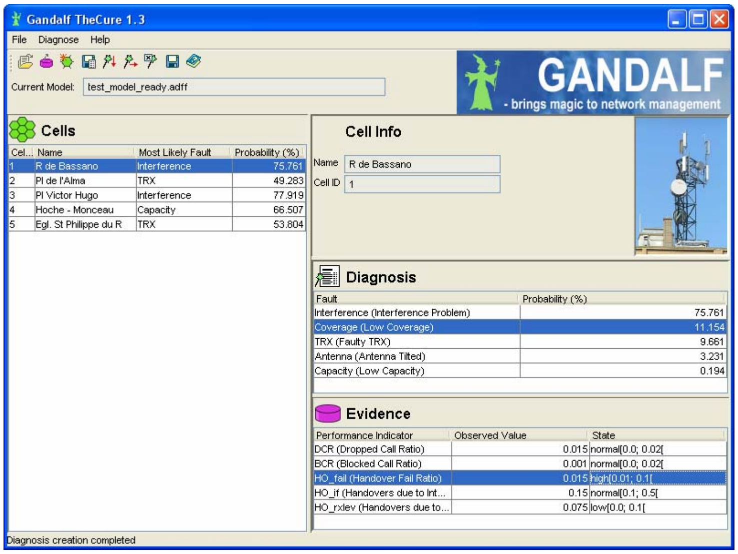

Figure 2. The execution tool known as "TheCure"

2.2 Toolset Functionality Listing

This section presents the complete list of functionality of the developed toolset.

2.2.1 Execution Tool: TheCure

- TheCure loads a model in the ADFF format (ADFF = Auto-Diagnosis File Format, an XML format provided by Wirtek (previously Moltsen Intelligent Software) enabling the specification of auto-diagnosis models)
- TheCure reads inputdata from a set of cells
- TheCure uses the loaded model and the input data to compute a diagnosis consisting of an ordered list of fault causes. The list is ordered by the probability of the fault causes
- TheCure presents diagnosis, inputdata, probabilities, and other relevant data in a user-friendly graphical user interface
- TheCure enables the user to store results in a text-based format (tabular-separated columns) which can be read by MS Excel and other tools that the user might choose for reporting.

2.2.2 Modelling Tool: Knowledge Builder

- Knowledge Builder enables the user to construct or edit a model which can afterwards be saved and re-loaded in the ADFF format
- Knowledge Builder enabls the user to specify a number of fault causes and a number of input variables (performance indicators).
- It is possible for each input variable to specify the fault cause this input variable is *relevant* for (the fault would trigger the input variable to get a different reading).
- It is possible to specify discrete intervals for continuous-range input variables, thus resulting in a finite set of states.
- It is possible to specify prior probabilities of fault causes and conditional probabilities of input variables given each relevant fault cause.
- Knowledge Builder can also learn the above-mentioned probabilities from data sets where both input data and true fault are known so that the user does not have to specify them
- Knowledge Builder enables the user to learn interval boundaries of continuous-range input variables from data sets, so that the user does not need to do this.
- Knowledge Builder also provides a *Wizard* which starts with an empty model and builds a model only by looking at the data set (finds all fault causes and input variables in the data, and then determines the state boundaries using the previous function).

3 Toolset Evaluation

Toolset evaluation has been carried out by the different trial persons responsible for auto-troubleshooting work in the partner organizations. Each person was given the questionnaire presented in Appendix A with no further instructions than what was given at the top of the questionnaire. It was ensured that the test persons did not see other people's responses before they had delivered own results.

The following three sections contain the feedback received from the three trial persons. The two first (France Telecom R&D and Telefonica I+D) were responsible for the D5.3a deliverable [2], whereas the last (University of Limerick) was a minor study not documented elsewhere in Gandalf context, but we considered the feedback to be significant and relevant for the project.

3.1 France Telecom R&D

Background Information	
User data (name, organization, email, phone):	**Jordi Triola Bosch** France Télécom Reseach & Development RESA/NET/REM 38-40 rue du Général Leclerc 92794 Issy les Moulineaux Cedex 9 France Phone: +33 1 45 29 66 14 jordi.triolabosch@orange-ft.com
Approximately, how much work have you done which was based on the toolset (months, weeks)?:	3 months
Shortly present the work which was based on the toolset:	We developed a troubleshooting model for a UMTS network including some fine tuning, and then we produced a set of results using data from a dynamic simulator.
Toolset/Concept Evaluation	
How well does the auto-troubleshooting toolset solve the practical problem of automated troubleshooting (range 1-5)?:	4
How important is the Bayesian technology for auto-troubleshooting (range 1-5, 1="Not needed", 5="Crucial")?:	4
How easy was it to learn how to use the toolset (range 1-5)?:	5
How easy was the toolset to use after learning (range 1-5)?:	5
How much time did it take to develop the model (including fine tuning, excluding time for you to learn about the domain)?:	One of the most important tasks is to well choose the pertinent PIs for each fault. (about 2 weeks in our case). We did some tests over two more weeks with this initial model, and then we were able to get a better accuracy by adding a few more performance indicators. In total 1 month.
How big is the model that you developed (number of faults and inputs/PIs)?:	8 faults and 10 PIs

For the domain where your model was built, approximately how many faults are needed to cover 90% of all troubleshooting cases (your best guess)?:	It strongly depends of the labels used to define the faults, but we would say about 10 faults.
In a scenario with a model having 20 faults and 20 inputs (PIs), how static is the model and how much work (mm/year) is needed to keep the model up-to-date?:	The more inputs are static over the time, the more the model will be static and less mm/year will be required to keep the model updated.
Assuming a scenario where an organization is doing troubleshooting manually, approximately how much of time spent on diagnostics will the auto-troubleshooting toolset be able to save (percentage)?	It's difficult to say, and it strongly depends of the diagnosis accuracy that the model has been able to reach. But if the model is quite well calibrated about 60% of time on diagnostics will be saved. Anyway, it will always be necessary a quick manual validation.
Do you have quantitative results that shows the performance of the toolset?:	*FT simulator results:* Setup 1: True fault was listed first by TheCure: 70.2% True fault was first or second by TheCure: 88.2% Setup 2: True fault was listed first by TheCure: 48.6% True fault was first or second by TheCure 71.4%
What do you think is the most important thing about the developed auto-troubleshooting toolset?:	It's very easy to use and it performs diagnosis results very quickly.
Additional Information	
Other comments or evaluation feedback not covered above?:	The Graphical User Interface is quite convivial.

3.2 Telefonica I+D

Background Information	
User data (name, organization, email, phone):	***Beatriz Solana*** ***TELEFONICA I+D*** ***Emilio Vargas,6*** ***28043 Madrid*** ***Spain*** ***Email: solana@tid.es*** ***Phone: +34 91 337 11 33***
Approximately, how much work have you done which was based on the toolset (months, weeks)?:	***2 months***
Shortly present the work which was based on the toolset:	***We developed a troubleshooting model for a UMTS network, and then we produced a set of results using data from a real network.***
Toolset/Concept Evaluation	
How well does the auto-troubleshooting toolset solve the practical problem of automated troubleshooting (range 1-5)?:	***4***
How important is the Bayesian technology for auto-troubleshooting (range 1-5, 1="Not needed", 5="Crucial")?:	***3***
How easy was it to learn how to use the toolset (range 1-5)?:	***5***
How easy was the toolset to use after learning (range 1-5)?:	***5***

How much time did it take to develop the model (including fine tuning, excluding time for you to learn about the domain)?:	***The initial model was made in a few days. But, we did many tests over three or four weeks, and then we were able to get a better accuracy by changing a few performance indicators. In total 1 month.***
How big is the model that you developed (number of faults and inputs/PIs)?:	***6 faults and 17 PIs***
For the domain where your model was built, approximately how many faults are needed to cover 90% of all troubleshooting cases (your best guess)?:	***With our model 6 faults are required.***
In a scenario with a model having 20 faults and 20 inputs (PIs), how static is the model and how much work (mm/year) is needed to keep the model up-to-date?:	***The model is very dynamic and needs at least 6 mm/year to keep it up-to-date.***
Assuming a scenario where an organization is doing troubleshooting manually, approximately how much of time spent on <u>diagnostics</u> will the auto-troubleshooting toolset be able to save (percentage)?	***Once performed the model: 75%.***
Do you have quantitative results that shows the performance of the toolset?:	***True fault was listed first by TheCure: 88%*** ***True fault was first or second by TheCure: 99%***
What do you think is the most important thing about the developed auto-troubleshooting toolset?:	***Once performed the model, you can find out the problem faster than manually. With the estimation, you will know how to solve the problem.***
Additional Information	
Other comments or evaluation feedback not covered above?:	***About Knowledge Builder, many input data are needed to get a good result, and this can turn out to be very costly.***

3.3 University of Limerick

Background Information	
User data (name, organization, email, phone):	Carlos de Antonio Ramos Student doing my graduate thesis. University of Limerick Email: 0633445@student.ul.ie Phone: +45 9635 6125
Approximately, how much work have you done which was based on the toolset (months, weeks)?:	4 weeks
Shortly present the work which was based on the toolset:	*At the moment, I use the toolset for some easy examples to understand how it works*
Toolset/Concept Evaluation	
How well does the auto-troubleshooting toolset solve the practical problem of automated troubleshooting (range 1-5)?:	4
How important is the Bayesian technology for auto-troubleshooting (range 1-5, 1=”Not needed”, 5=”Crucial”)?:	*4*
How easy was it to learn how to use the toolset (range 1-5)?:	4
How easy was the toolset to use after	5

learning (range 1-5)?:	
How much time did it take to develop the model (including fine tuning, excluding time for you to learn about the domain)?:	I am sorry I only did tests simulations
How big is the model that you developed (number of faults and inputs/PIs)?:	3 faults and 3 PIs
For the domain where your model was built, approximately how many faults are needed to cover 90% of all troubleshooting cases (your best guess)?:	10
In a scenario with a model having 20 faults and 20 inputs (PIs), how static is the model and how much work (mm/year) is needed to keep the model up-to-date?:	The model is quite static and less than 1 mm/year is required to keep it updated.
Assuming a scenario where an organization is doing troubleshooting manually, approximately how much of time spent on diagnostics will the auto-troubleshooting toolset be able to save (percentage)?	70.00%
Do you have quantitative results that shows the performance of the toolset?:	*No yet.*
What do you think is the most important thing about the developed auto-troubleshooting toolset?:	The time that organization can save using autotroubleshooting. User-friendliness
Additional Information	
Other comments or evaluation feedback not covered above?:	

4 Conclusions

A demonstration toolset for automated troubleshooting has been developed and validated thoroughly both by quantitative results (documented by [2]) and by feedback questionnaires documented in this deliverable.

One of the most interesting findings of presented toolset evaluation is that all trial persons agree that the estimated savings of using an auto-troubleshooting tool for diagnostics is greater than the 50% savings estimated in the Techno-Economical Study [3]. In fact, the three trial persons have estimated between 60% and 75% savings. This means that the tool will provide great benefit to the operator integrating this type of tool in their organization.

There is some uncertainty on the maintenance effort. Telefonica I+D indicates a relatively high maintenance effort and lack of available data. University of Limerick points towards low maintenance, but their model was also much smaller. France Telecom R&D does not provide a quantitative answer to this question. To conclude, this is probably the most critical issue to be targeted for improvements of the concept (one could aim to develop a special case databases which should then be supported by proper processes in the operator organization – maybe something to solved by another research project?).

Another important observation is that the toolset is very easy to both use and to learn to use. All three feedback forms agreed that the toolset was “Very good!” when asked how easy the toolset was to use. Two out of the three also had “Very good!” in how easy it was to use the tool. The last one was “Above average/above expectations”. This is equally interesting to the operator who is considering to use the tools because there is always a cost involved in installing new software to support the organization. In this case, we can argue that the cost is not that significant.

5 References

[1] Gandalf Deliverable 5.2, *Software package for self-tuning and auto-diagnosis*, December 2006

[2] Gandalf Deliverable 5.3a, *Automating Diagnosis in Troubleshooting*, December 2006

[3] Gandalf Deliverable 2.2, *Techno-Economical Study*, March 2007

Appendix A: Gandalf TheCure Evaluation Form

The following was the exact questionnaire as presented to the trial person:

Purpose:
The purpose of this evaluation form is to gather user experiences about the developed auto-troubleshooting toolset, known as TheCure (the toolset contains an execution tool named *TheCure* and a tool for model construction named *Knowledge Builder*).

Capturing knowledge and experiences from early user will be very important for dissemination activities since it will be the main communication link between the project and potential future users in the telecom industry.

Instructions to fill in the form:
Please fill in your inputs in the second (right) column in the form below. The input should be short and precise and focus on tool/concept evaluation (not evaluation of your work and results). Be aware, that the reader of the form will be externals to the project, so only reveal information that you would also be willing to put in a conference paper (although it is not intended that the form shall not be published to a broad audience).

Most questions already have example data. Simply delete this and type your own data there.

For the quantitative questions (range 1-5), this should be the meaning of the values:

1: Poor
2: Below average/below expectations
3: As expected
4: Above average/above expectations
5: Very good!

Form:

Background Information	
User data (name, organization, email, phone):	*Example:* Lars Moltsen Moltsen Intelligent Software Niels Jernes Vej 10 9220 Aalborg Ø Denmark Email: lars.moltsen@moltsen.com Phone: +45 9635 6125
Approximately, how much work have you done which was based on the toolset (months, weeks)?:	*Example:* 3,5 months
Shortly present the work which was based on the toolset:	*Example:* We developed a troubleshooting model for a GSM network including some fine tuning, and then we produced a set of results using data from a real network.
Toolset/Concept Evaluation	
How well does the auto-troubleshooting toolset solve the practical problem of automated troubleshooting (range 1-5)?:	*Example:* 3

How important is the Bayesian technology for auto-troubleshooting (range 1-5, 1="Not needed", 5="Crucial")?:	*Example:* 3
How easy was it to learn how to use the toolset (range 1-5)?:	*Example:* 3
How easy was the toolset to use after learning (range 1-5)?:	*Example:* 3
How much time did it take to develop the model (including fine tuning, excluding time for you to learn about the domain)?:	*Example:* The initial model was made in a few days. We did some tests over two weeks, and then we were able to get a better accuracy by adding a few more performance indicators. In total 2 weeks.
How big is the model that you developed (number of faults and inputs/PIs)?:	*Example:* 10 faults and 8 PIs
For the domain where your model was built, approximately how many faults are needed to cover 90% of all troubleshooting cases (your best guess)?:	
In a scenario with a model having 20 faults and 20 inputs (PIs), how static is the model and how much work (mm/year) is needed to keep the model up-to-date?:	*Example:* The model is quite static and less than 1 mm/year is required to keep it updated. *or* The model is very dynamic and needs at least 6 mm/year to keep it up-to-date.
Assuming a scenario where an organization is doing troubleshooting manually, approximately how much of time spent on diagnostics will the auto-troubleshooting toolset be able to save (percentage)?	
Do you have quantitative results that shows the performance of the toolset?:	*Example (the FT simulator results):* Setup 1: True fault was listed first by TheCure: 74.5% True fault was first or second by TheCure: 91.8% Setup 2: True fault was listed first by TheCure: 57.1% True fault was first or second by TheCure 88.6%
What do you think is the most important thing about the developed auto-troubleshooting toolset?:	
Additional Information	
Other comments or evaluation feedback not covered above?:	