Using GQM Hypothesis Restriction to Infer Bayesian Network Testing

MSc. Denis Ávila Montini  
*Instituto Tecnológico de Aeronáutica - ITA*

Eng. Felipe Rafael Motta Cardoso  
*Instituto Tecnológico de Aeronáutica - ITA*

Prof. Francisco Supino Marcondes  
*Instituto Tecnológico de Aeronáutica - ITA*

Dsc. Paulo Marcelo Tasinaffo  
*Instituto Tecnológico de Aeronáutica - ITA*

Dsc. Adilson Marques da Cunha  
*Instituto Tecnológico de Aeronáutica - ITA*

Abstract

By definition, the scope of a Bayesian Network uses a complementary technique to restrict the modeling reach. In this paper, the used restriction technique was the Goals, Questions, and Metrics (GQM). The hypothesis to be tested relates cause and effect conditional probabilities in a software test phase of a manufacturing production line. The Bayesian Network concept is related to the specific concept of a Directed Non Cyclic Graph (DNCG), where each one of its nodes represents a random discrete variable and is illustrated by directed arcs of cause and effect relationships between variables. A Bayesian Network is a graphical artifact which restricts problems, incorporating data structures. The major contributions of this paper are conceptualization and implementation of a methodology for using a GQM hypothesis restriction to infer Bayesian network testing with the Netica Bayesian Networks® computer software.

**Keywords:** Software house, manufacture cell, GQM, Final Inspection (FI), test, and Bayesian network.

1. Introduction

The conceptual elements of Bayesian Networks (BN) [1] are presented in this paper in the context of software house application test phase theory and software process using the Netica software tool [2]. The BN identification was restricted through the use of the Goals, Questions, and Metrics (GQM) technique [3].

A Directed Non Cyclic Graph (DNCG), as a specific case study of the Graphical Network Bayesian Theory, represents a capacity that has been explored on practical studies. This uncertainty area of probability has been used in different subjects like software engineering and production lines due to its elegance of solution when codified using BN.

On human manufacturing activities, forecasting production behavior notions does exist, as well as probability applications which make possible statistical manipulations.

This paper describes an investigation of probability by using the Bayes Theorem and GQM to define BN. It presents the conceptualization and implementation of a methodology for using GQM hypothesis restriction to infer BN testing with the Netica Bayesian Networks® computer software. In order to symbolically represent information systems about a phenomenon knowledge [4] [5], BN may use graphical tree projects.

A real world modeling is based on mathematical process formalization to incorporate real world elements. To be validated, these relationships allow true mathematical mechanisms to study a joint distribution of probabilities involving random variables. Selected natural variables with stochastics dependencies are organized by facts and specific rules gotten from expert systems [6], resulting in an expert database for BN.

The organization of these concept base formations has involved probabilistic expert systems leading to the discovery of knowledge from classified database.

2. Methodology

The conceptualization and implementation of the GQM-BN integration methodology are the major contributions of this paper. This methodology is comprised of the steps shown in Table 1.

Table 1. The GQM – BN integration methodology

<table>
<thead>
<tr>
<th>Steps</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. GQM identification.</td>
<td>To define the GQM application.</td>
</tr>
<tr>
<td>2. Restricting BN Modeling.</td>
<td>To restrict BN Modeling with GQM requirements.</td>
</tr>
<tr>
<td>3. Restricting BN by using the GQM test.</td>
<td>To implement the GQM Model in the Netica software tool.</td>
</tr>
<tr>
<td>4. Probabilistic Inferencing.</td>
<td>To accomplish probabilistic inferences for BN.</td>
</tr>
<tr>
<td>5. PCP elaboration.</td>
<td>To elaborate PCP for the test phase.</td>
</tr>
<tr>
<td>6. Process customization.</td>
<td>To use the GQM-BN application.</td>
</tr>
<tr>
<td>7. GQM-BN analysis.</td>
<td>To analyze results from the GQM-BN application.</td>
</tr>
<tr>
<td>8. Decision making.</td>
<td>To make decisions regarding tested software products.</td>
</tr>
<tr>
<td></td>
<td>1.1 - If test pass, them go to the step 9.</td>
</tr>
<tr>
<td></td>
<td>1.2 If the test fails, go back to the</td>
</tr>
</tbody>
</table>
2.1 The QGM identification step

Table 2. The QGM technique case study model

<table>
<thead>
<tr>
<th>Business Goal</th>
<th>To assure that all known defects were corrected before the software is released.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement</td>
<td>To remove all known defects in the software testing process.</td>
</tr>
<tr>
<td>Questions</td>
<td>1-2 - What is the number of defects of the system? 3-4 How many units, integrated and system defects are currently there? 5-6 What are the held coverage and planned testing? 7-8 What are the number of requirements directly designed and tested?</td>
</tr>
<tr>
<td>Metrics</td>
<td>1 - Number of defects by quantitative state 2 - Number of defects by qualitative state 3 - Number of quantitative defects 4 - Number of qualitative defects 5 - Number of quantitative held coverage test requirements 6 - Number of qualitative held coverage test requirements 7 - Number of quantitative test requirements 8 - Number of qualitative test requirements</td>
</tr>
</tbody>
</table>

The QGM identification was obtained by applying the GQM technique [3] and it can be visualized in a case study. Table 3 presents GQM technique application results. It was applied to restrict the test phase.

Following it is presented an explanation on how Table 3 GQM was fulfilled.

Business Goal - To ensure that all known defects are corrected before software is released for use.

Measurement Business - To remove all known defects from the software testing process.

Questions - "1-2 - What is the number of defects of the system?"; "3-4 How many units, integrated and system defects are currently there?"; "5-6 What are the held coverage and planned testing?"; and "7-8 What are the number of requirements directly designed and tested?".

Metrics - "1 - Number of defects by quantitative state"; "2 - Number of defects by qualitative state"; "3 - Number of quantitative defects"; "4 - Number of qualitative defects"; "5 - Number of quantitative held coverage test requirements"; "6 - Number of qualitative held coverage test requirements"; "7 - Number of quantitative test requirements"; and "8 - Number of qualitative test requirements".

2.2 The restricting BN modeling step

In order to implement the restricting BN modeling, it was necessary to research GQM methods to develop Bayes Theory elements. To do this, it was adopted a GQM test phase with BN related to Graph Theory. These conditions were adopted to represent nodes and arcs in variables.

Each node and arc pair represented a probabilistic dependency between a quantitative and a qualitative variable association, as shown in Figure 1.

A used information system organization can be schematically visualized using GQM and BN [8], both shown in Table 1 and Figure 1. Each paired questions 1-2, 3-4, 5-6, and 7-8 had two answer metrics, one quantitative and other qualitative [3] [7] to be implemented within an information system application.

The DNCG model [9] had variables represented by nodes and arcs connecting the quantitative and qualitative variables defined by GQM, as shown in Figure 1.

According to Table 3 and Figure 1, the goal node was considered the root or exit node. Metrics nodes from 1 to 8 were considered the leaves, entrances, or primitive nodes. This paper presents how one BN was proposed, defined, and constructed for a decision making process based upon inference rules. The BN presented in Figure 1 allowed a product diagnostic during the test phase of a System Development Life Cycle (SDLC) using the Netica software tool [2].

2.3 The restricting BN by using GQM test step

The use of a DNCG for dependencies relationship has represented a powerful tool for verification process and knowledge acquisition [17].

A restricted universe for GQM helped to define a set of discrete variables to be shaped in BN. When shaping a
BN using the GQM technique some interest areas of relationship needed to be defined. In this case, it was used the Software Engineering Project and the Planning and Control Production (PCP) areas.

This representation set was comprised of local probabilities within tree structures. This allowed an independent conditional inference set for a global distribution construction from local distributions.

The restricted shape measures presented in Figure 1 together with rules “2.” and “3.” from Table 3 were used to build the software application shown in Figure 2. Figure 2 represents a snapshot of a real time embedded software component in the inspection test phase. The first action to construct a BN was deciding which variables had to be used, and how many states the model would have.

After that, the inferred result was: “To assure that all known defects were corrected before the software is released”. A second action was defining for each GQM some questions to allow the building of a tree by applying the BN Theorem. Figure 2 presents the results of this action.

### Table 3. GQM inference rules for the BN testing phase

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>What is the absolute number of system defects? with Worst, Bad, Reasonable and Good;</td>
</tr>
<tr>
<td>1-1-1</td>
<td>Quantitative: with Worst states, Bad, Reasonable and Good;</td>
</tr>
<tr>
<td>1-1-1-1</td>
<td>Data Input “0” or “1”: with states (0 to 1), (1 to 2), (3 to 4), (5 to 6), (7 to 8), or (9 to 10).</td>
</tr>
<tr>
<td>1-1-2</td>
<td>Qualitative: with Worst states, Bad, Reasonable and Good;</td>
</tr>
<tr>
<td>1-1-2-1</td>
<td>Data Input “0” or “2”: with states (0 to 1), (1 to 2), (3 to 4), (5 to 6), (7 to 8), or (9 to 10).</td>
</tr>
<tr>
<td>1-2</td>
<td>How many Unitary, Integrated, and System defects are there actually in the system? with Worst, Bad, Reasonable and Good;</td>
</tr>
<tr>
<td>1-2-1</td>
<td>Quantitative: with Worst state, Bad, Reasonable and Good;</td>
</tr>
<tr>
<td>1-2-1-1</td>
<td>Data Input “0” or “1”: with states (0 to 1), (1 to 2), (3 to 4), (5 to 6), (7 to 8), or (9 to 10).</td>
</tr>
<tr>
<td>1-2-2</td>
<td>Qualitative: with Worst state, Bad, Reasonable and Good;</td>
</tr>
<tr>
<td>1-2-2-1</td>
<td>Data Input “0” or “2”: with states (0 to 1), (1 to 2), (3 to 4), (5 to 6), (7 to 8), or (9 to 10).</td>
</tr>
<tr>
<td>1-3</td>
<td>What are the relative numbers of requirements planned and performed in the test phase? with Worst, Bad, Reasonable and Good;</td>
</tr>
<tr>
<td>1-3-1</td>
<td>Quantitative: with Worst state, Bad, Reasonable and Good;</td>
</tr>
<tr>
<td>1-3-1-1</td>
<td>Data Input “0” or “1”: with states (0 to 1), (1 to 2), (3 to 4), (5 to 6), (7 to 8), or (9 to 10).</td>
</tr>
<tr>
<td>1-3-2</td>
<td>Qualitative: with Worst state, Bad, Reasonable and Good;</td>
</tr>
<tr>
<td>1-3-2-1</td>
<td>Data Input “0” or “2”: with states (0 to 1), (1 to 2), (3 to 4), (5 to 6), (7 to 8), or (9 to 10).</td>
</tr>
<tr>
<td>1-4</td>
<td>What are the absolute numbers of requirements planned and performed in the test phase? with Pessimist states, Bad, Reasonable and Good;</td>
</tr>
<tr>
<td>1-4-1</td>
<td>Quantitative: with Worst state, Bad, Reasonable and Good;</td>
</tr>
<tr>
<td>1-4-1-1</td>
<td>Data Input 01: with states (0 to 1), (1 to 2), (3 to 4), (5 to 6), (7 to 8), or (9 to 10).</td>
</tr>
<tr>
<td>1-4-2</td>
<td>Qualitative: with Worst state, Bad, Reasonable and Good;</td>
</tr>
<tr>
<td>1-4-2-1</td>
<td>Data Input 02: with states (0 to 1), (1 to 2), (3 to 4), (5 to 6), (7 to 8), or (9 to 10).</td>
</tr>
</tbody>
</table>

The third action was modeling process implementation to construct a BN through an expert system tool. The DNCG supported values and characteristics to insert for any node respecting the independence conditional theory.

By implementing all these restrictions, a new BN structure supported the Bayes Theorem. Under multiple hypothesis and multiple independent evidences data were gotten from real world projects. The fourth and final action was defining a local distribution for each state, as shown in Figure 2.

### 2.4 The probabilistic inferencing step

A probability interpretation is a degree of belief called subjective interpretation or Bayesian. In Bayesian interpretation, probability was always dependent on knowledge from which it provides the probability. In an actual expert system, a person can evaluate the probability based upon information assumed to be truth.

For an inference probabilistic planning, process formulation and efficient control were necessary to identify different reasoning forms.
The Netica software tool has a database to register all the inferences made by team members. Software professionals need a BN mental map, in order to decide during the test phase. Figure 3 presents the resulting database from this applied analytical process.

2.5 The PCP elaboration step

To acquire an empirical knowledge about PCP projects, the PCP was applied to manufacturing cells [10, 11, 12, 13, 14, 15, and 16] in some different software houses to build GQM-BN. It was necessary some knowledge for software dimension, according to some available evidences based upon software engineering good practices.

2.6 The process customization step

A GQM-BN customization process was performed when a BN was connected to a historical database. Team members updated the form constructed to adapt the current situation and brought the up-to-date estimated probabilities from a stored data called Adapted Bayesian Networks (ABN).

This capability has offered the possibility to implement Knowledge Database Discovery (KDD) techniques.

2.7 The GQM-BN analysis step

A GQM-BN analysis also was represented as qualitative research rules applied to Goals, Questions, and Metrics with the purpose of modeling human reasoning, for a computational development system, and a rational decision taking environment [3].

Sets of pointers were modeled in functions. The GQM held different types of analysis. For this reason, the use of BN was tested and mapped within an interval value scaling from “0” to “1”, in order to indicate its degree of relevance to this set.

Also inference rules were established, normally being extracted from human experts or numerical values. From signals, symptoms and complaints for patient inference process determining how rules were activated and combined.

2.8 The decision making step

The building of a decision support system for a project management had the intention of becoming a tool to help solving contingency problems and analyzing the software product quality. The aiming here was to discipline human beings operations in order to diminish resultant residual software defects during the construction process.

In this restricting scenario of operating research, the infused defects during the construction phase were treated through KDD techniques, established in this case study by using BN modeling [19]. Some possible solutions for this hypothesis have considered the existence of some available foundations in Artificial Intelligence applications and techniques to support the decision taking process [20].

2.9 The database storing step

Information resulting from data processing, manipulation, and organization has represented the added value of knowledge to end users [17]. This empirical knowledge was systemized through a constant flow of information stored in historical databases.

Knowledge based on evidences was obtained from accumulated evidences throughout the time, and it was summarized by adjusted statistical processes. Formal knowledge base on engineering literature, available in periodics, books, and other forms of storage still have included deduction processes [5].

The historical base for data calibration using structured and non structured information was obtained from statistical techniques and from analysts’ errors probabilistic rates [18].

The restriction scope was according with business objectives and form with that programming step followed by company lines of direction.

Moreover, a contract that states the basis of a project technique can be shaped through GQM production rules and submitted to Bayesians inference for management information systems formation [5].

With these theories in mind, the composition considers Bayes Theory in multiple hypotheses treatment and multiple restricted evidences for an organizational delimiter. This boarding is adherent to probabilistic independence concept between distinct events, in which data on an event do not bring additional information to
another event. Thus, given that an event has occurred, this does not bring information for event 2. This however is affected by eventual occurrences in known structure, correcting probability, from each new register of success studied phenomenon gotten.

3. Applications

A BN application was shaped for control and plan the software production lines knowledge, and to get data from this behavior. A natural consequence of this register was to improving the project estimation quality in other projects.

The control of quality and the decision making application with GQM-BN were solved to a product homologation. Its application was used to the knowledge acquisitions. The GQM-BN had many applications in software production engineering, software management, function points, and so on.

The Netica tool application was used to describe discrete variables and data insertion in a specific node, when a phenomenon occurs. The Netica software also made possible the use of the decision theory with GQM-BN model. This GQM-BN model has included more than 70 test cases distributed in 15 Bayesians nodes.

In order to check this application, a component test case was built in a Real Time Embedded Software (RTES) developed in C++. The GQM-BN software application represents how the manager has observed these tests during the User Acceptance Test (UAT) phase, shown in Figure 2.

Each decision node has permitted users actions to modify the model during the actual action and made possible to verify if the test project planning forecast was corrected dimensioned.

4. Result

A theoretical study on GQM-BN to understand its basic concepts and practical application was done in this research. This GQM-BN application has allowed a scope restriction for GQM inference hypothesis testing.

The major results of this paper were: the conceptualization of the GQM-BN methodology; the implementation of the GQM-BN application; and the construction of a probabilistic calibrated table for Bayesians Inferences of a production line using Real Time Embedded Software (RTES).

A GQM-BN application tool was built with high degree of usability. It allowed the visualization of a Bayesian Graph, as shown in Figure 2, as well as a Probabilistic and Decision Table, as show in Figure 3. From both tables, inferences can be quickly calculated by the GQM-BN application [21].

During the Verification and Validation [17] process, a GQM-BN model was also evidenced. This model had a distinct format to deal with human interaction. However, the obtained result needs to be put in a GQM-BN system by each team member.

The GQM-BN application allowed to find “3” errors in unitary tests with high severity. In the systems test it was found “0” error, and in the integrated test “0” error was found too, during the two homologation test cycles.

According to theses evidences the team members could update the GQM-BN application. The results consolidation had provide the identification of “3” errors with high severity in “70” possible cases.

This paper presented a BN implementation in an expert process for RTES. A risk probability evolution graph was obtained with evidences from this process, as Figure 4 shown.

5. Conclusion

This paper has summarized the conceptualization and implementation of a methodology for using GQM hypothesis restriction to infer Bayesian Network (BN) testing with the Netica Bayesian Networks ® computer software.

Figure 4 presents the probability evolution to control errors found in computer software components developed in a laboratory. It shows the monitoring of the success probability according to the errors found throughout the time in the test phase.

![Figure 4. Evolution of risk probability, based in errors found in tests: Unitary, Integrated and System.](image)

Data were collected through quality inspections activities. These routine activities [18] have used probabilistic trends to find out errors from “0” to “1” in the test phase for user acceptance.

To a software house, with the union of GQM techniques proposed by Basili with the BN it was possible to apply an information system to hold multiple hypotheses and multiple evidences.

Some divergences have occurred due to the model, because a lack of refinement does exist to reproduce reality. The studied model had considered a great number
of test errors used to support a decision making during the verification and validation process phase.

This control system can allow the construction of IA expert systems, by reproducing knowledge structures, regarding manufacturing lines of production cells [10] behavior.

The use of expert tools as Netica facilitates Bayesian theory implementation, and the GQM utilization brought up the possibility to get in contact with concepts not very seen yet in the computation arena.

6. Recommendations

In order to continue this research, it is recommended:

1 - The utilization of the GQM concept in other domains;

2 - The systematic reuse of components on larger scale systems (scalability); and

3 – The application of the GQM-BN methodology on software houses and factories to scale up profits for RTES production lines [13, 17, 18, 19, 20, and 21].

For future work, it is suggested the results of this research be applied to different production lines to assess its feasibility.

7. Acknowledgements

Authors of this paper would like to acknowledge:

1 – The Brazilian Aeronautics Institute of Technology (ITA) for the opportunity of providing the course CE-235 Real-time Embedded Systems for Projects involving PBL;

2 – The GPES for providing its research environment;

3 – The FCMF for its software engineering research infrastructure; and

4 – The Tata Consultancy Services (TCS) as well, for its partnerships with productive sector companies.

8. References


[21] Tata Consultancy Services – An Enterprise from India acting on the consultancy and software factory segment in Brazil, since 2003.