

DRAFT

full paper published in:
**Lecture Notes in Computer Science,
Volume 5082, Berlin/Heidelberg, 2008, pp. 26-39**

**Proceedings of the 2nd IFIP Central and East European
Conference on Software Engineering Techniques CEE-SET,
Poznan, Poland, 2007, pp 1-15**

Paper published in the journal and presented at the conference

Extending GQM by Argument Structures

Łukasz Cyra, Janusz Górski

Gdańsk University of Technology, Department of Software Engineering,
Narutowicza 11/12, 80-952 Gdańsk, Poland
lukasz.cyra@eti.pg.gda.p, jango@pg.gda.pl,

Abstract. Effective methods for metrics definition are of particular importance, as measurement mechanisms are indispensable in virtually any engineering discipline. The paper describes how the well known Goal-Question-Metric (GQM) method of systematic metrics derivation from measurement goals can be extended by applying argument structures. The proposed approach is called Goal-Argument-Metric (GAM). The general ideas of GQM and GAM are briefly introduced and they are followed by the comparison of the two approaches. Next, the Trust-IT framework is described – it is used to develop argument structures in GAM. Then a case study of application of GAM is presented. The case study concerns derivation of metrics and direct measurements with the objective to assess effectiveness of Standards Conformity Framework (SCF), which is currently under development. In conclusion, early experience with GAM is presented and more information about on-going research on argument structures is given.

Keywords: GAM, GQM, Trust-IT, Trust case, Standards Conformity Framework, Measurement plan, metrics.

1 Introduction

Measurement mechanisms provide feedback that helps in evaluation of the actions undertaken and their results. However, identifying the scope of raw data to be collected and the metrics to be calculated from these data in order to achieve a particular measurement goal is a difficult and error-prone task. Selection of appropriate metrics which fit for the purpose and which do not generate unnecessary costs is a challenge. Implementation of the raw data collection process (which often needs non-trivial involvement of human effort) may be highly resource consuming. Collecting insufficient or excessive data and metrics can be frustrating and can

undermine the whole measurements initiative. Therefore, it is of primary importance that measurement plans make evident the objectives and the scope of collected data and the resulting metrics.

The above problems provided strong motivation for the research towards development of effective and efficient methodologies supporting systematic metrics derivation. An example is Goal-Question-Metric (GQM) [1, 9], which is a well known methodology targeted at defining measurement plans.

In this paper we propose a modification of GQM which we call Goal-Argument-Metric (GAM). The purpose is to provide solutions to some problems arising while using GQM and in particular to provide better support for the identification and maintenance of the relationship between the measurement goals and the related metrics. The basic innovation concerns applying argument structures for stepwise refinement of the measurement goals into metrics and direct measurements and by maintaining the arguments in the easily readable and accessible form with the help of our TCT tool.

The paper first introduces GQM and GAM and compares the two approaches. Then it introduces the Trust-IT framework [3-7], which we use for expressing and maintaining argument structures in GAM. The applicability of GAM is then illustrated by a case study. In this case study we refer to the problem of assessment of the effectiveness of Standards Conformity Framework (SCF) [2, 7]. SCF, which is presently under development, is a part of Trust-IT and its objective is to support processes of achieving and assessing compliance with standards.

In conclusion we summarise our contribution and present plans for further research.

2 GQM

The Goal-Question-Metric (GQM) methodology was originally developed by V. Basili and D. Weiss and then significantly extended by D. Rombach. It is a practical methodology which helps in systematic derivation of measurement plans. GQM is well documented, for a thorough description see e.g. [1, 9]. Many other sources are also available on the Internet. The idea of GQM is graphically presented in Fig. 1.

GQM proceeds top-down, starting with the definition of an abstract measurement goal, which explicitly represents the measurement intent.

Then, referring to this goal, several questions are defined which brake the problem into more manageable chunks. The questions are defined in such a way that obtaining the answers to the questions leads to the achievement of the measurement goal. This step is the most difficult one, as deciding about the level of abstraction of the questions is by no means a trivial task. It is easy to make the questions too abstract or too detailed. In both cases, difficult problems related to identifying the relationship between the questions and the collected data and metrics, or the problems related to interpreting answers to the questions in the context of the measurement goal, may arise [9]. Therefore, a substantial experience is usually necessary before one can

effectively apply GQM. This causes that the implementation of GQM requires a significant initial effort [9].

In the next step, based on the questions metrics are defined, which provide quantitative information then treated as answers to the questions. Finally, at the lowest decomposition layer, direct measurements are defined which provide the data necessary to calculate the metrics (see Fig.1).

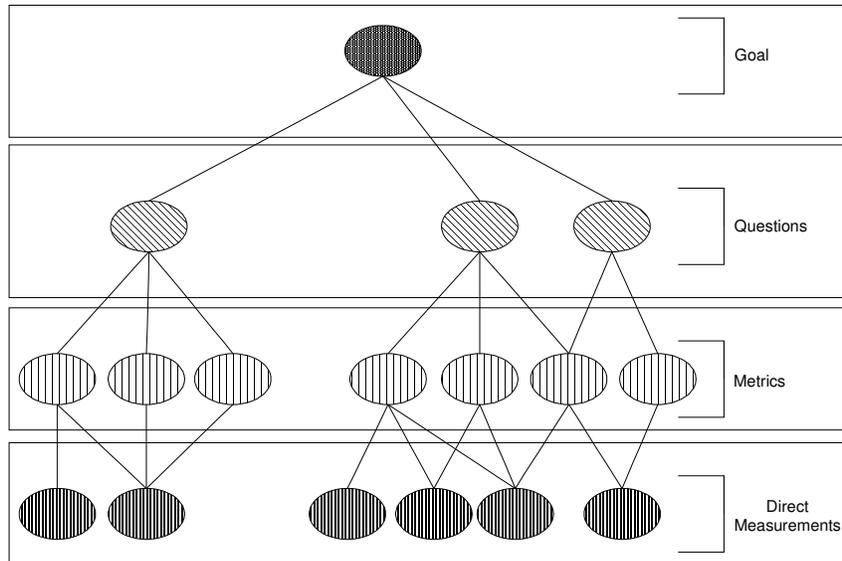


Fig. 1. GQM paradigm. Defining measurement goals (abstract level) refining them into questions (operational level), deriving metrics (quantitative information) and direct measurements.

As the problem of defining ‘good’ questions is not easy and had no obvious solution, some additional steps have been proposed with the intention to bring more precision to GQM. For instance, templates for defining the measurement goal have been introduced and supported by different types of models providing additional explanatory information. The template requires that the goal is defined in a structured way, including: the object of study, the purpose, the quality focus, the viewpoint and the context. The structure of such a definition is as follows:

Analyze	<the object of study>
for the purpose of	< the purpose>
with respect to	<the quality focus>
from the viewpoint	<the viewpoint>
in the context of	<the context>

It has been confirmed by practical experience that the increased precision and clarity in the goal definition positively influences suitability and usefulness of the measures derived with the help of GQM [1].

Additionally, GQM can be supported by models of different types with the intention to better represent the domain knowledge. It has been suggested that descriptive, evaluation, and predictive models are applied to help in 'grounding' the abstract attributes, defining relationships between objects of different types, and making predictions.

GQM has evolved in time into its model-based variant, which explicitly considers models of processes and products. However, the basic idea is still the same: to derive metrics from goals using the three-step top-down procedure inspired by Fig.1.

3 GAM

Following the main idea of GQM, GAM is a goal-oriented methodology for defining measurement plans. It differs, however, in the way the metrics and direct data measurements are derived from the goals. Instead of using partial solutions like templates and models, GAM provides seamless way of increasing precision in the whole process of metrics derivation.

In GAM, the goals and sub-goals are represented as claims and then the analysis focuses on identifying which data and which properties of the data (further sub-goals) are needed to demonstrate these claims.

The starting point is a claim postulating that the overall measurement goal has been achieved. Then the claim is justified by giving an argument which supports the claim. The argument can refer to other claims (about certain postulated properties) representing more manageable components of the problem. The inference rule used in the argument is stated explicitly showing the assumed argumentation strategy. If this rule is not self evident, another argument may be needed to demonstrate the validity of the inference rule. Such an argument can refer to the context of the goal (for instance, to argue the completeness of the evidence considered in the inference rule).

The procedure of decomposing claims into sub-claims is then repeated iteratively until it is possible to argue the leaf claims by directly referring to values of certain metrics. In such cases, the claims are supported by assertions on prospective values of such metrics (i.e. the assertions about metrics), as shown in Fig.2.

In the next step, the assertions are used to build a list of metrics, which is usually a trivial task. Finally, direct measurements are derived from the metrics to define the scope of raw data to be collected.

From the argument structure which links the data and metrics with the measurement objective, it is straightforward to implement the bottom-up process that gathers the raw data (by means of direct measurements) and aggregates them into metrics. If the obtained values meet the criteria given by the assertions kept in the argument structure, the whole argument tree explicitly demonstrates that the initial goals have been met.

Fig.2 illustrates the GAM approach in a graphical form.

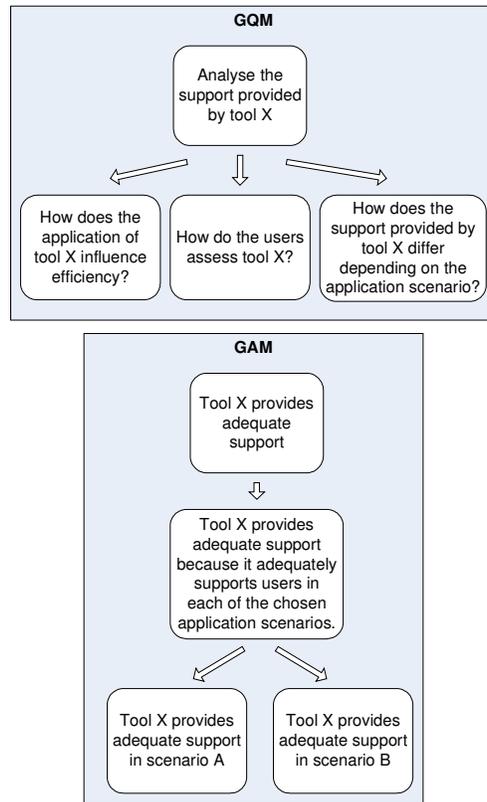


Fig. 3. Comparison of GQM and GAM. Defining a set of questions on the basis of the measurement goal and a set of sub-claims demonstrating the root claim.

The example presents a refinement of the measurement goal which is to assess the support provided by a tool X. In case of GQM, a set of questions is defined with the intention to cover all the aspects related to analysis of the support provided by tool X. Identification of such questions is not an easy task and the analyst has to constantly control the scope of the analysis. By contrast, in case of GAM the focus is on finding an argumentation strategy which demonstrates the adequate support offered by tool X. In the example, the strategy is by considering different application scenarios for X. Once the strategy has been chosen, the refinement into sub-claims is a natural consequence.

In GQM, choosing an appropriate level of abstraction for the questions is (according to [9]) a difficult task and a substantial experience in application of the method is needed. It is possible to use more than one level of questions to make the definition of “proper” questions easier. The relationship between the adjacent levels of questions is not explicit, which increases the difficulty in using the method. Some approaches to deal with this difficulty have been proposed, for instance, the interpretation models of different types [1].

GAM admits multiple levels of claims and does not restrict the user in this respect. At each level, the problem is broken into more manageable sub-problems and the relationship between the adjacent levels is explicitly established by giving the corresponding argument. The subsequent abstraction layers result naturally from the task of justifying the higher level claims by referring to the lower level ones. In order to create sound warrants for the arguments, the user of GAM is usually forced to refer to the information that is represented in the models foreseen in GQM, however, in this case it is simply a part of the argument development process and the scope of this information is easily controlled.

In our assessment, the most significant advantage of GAM is the introduction of argument strategies and warrants, which support the arguments. Considering what is necessary to justify a claim, finding an appropriate argumentation strategy and documenting those decisions provides for focusing the scope of the analysis and traceability of the results. Arguments make it evident whether the sub-components are necessary to support the claim and whether the decomposition is complete. Therefore, the questions like: 'Is it a complete set of questions which must be taken into account?' or 'Do I really need this question to support the goal?' do not appear in GAM.

Both methods are supported by advanced tools. For instance, in [8] a tool supporting GQM has been described. GAM is fully supported by the TCT tool [10] which is part of the Trust-IT framework. This framework is described in more detail in the subsequent sections.

5 Trust Cases

Development of arguments in GAM follows the approach defined in the Trust-IT framework. A part of the Trust-IT framework is the trust case language which provides means of expressing arguments [3, 4, 5, 7]. Similar languages have been used in the safety critical systems domain to express 'safety cases' – arguments justifying that a given system is adequately safe while considered in its target context. Trust cases differ from safety cases in several respects, for instance they can address broader (practically unlimited) scope of properties and do not have any particular restrictions on their structure and contents. We have already applied trust cases to analyse and justify different properties, including safety, security, privacy and others. Another interesting area of application of trust cases is assessing and demonstrating the compliance with standards, which we are presently investigating.

In GAM we represent argument structures as trust cases. Flexibility of the language and legibility of the arguments are two important factors which influenced this decision. Additional advantage is that trust cases are supported by an efficient Internet-enabled tool [10] which supports management and sharing of trust case structures.

Trust cases are composed of nodes of different types. The type of a node represents its role in demonstrating a certain statement. The basic logical component of a trust case is an argument composed of a *claim* to be justified (denoted **CL**), *evidence*

supporting the claim and an *inference rule* which shows how, on the basis of the evidence, the claimed property is achieved.

The evidence and the claim are connected using nodes of type *argument* (denoted **Arg**), which state the argumentation strategy. Apart from arguments also *counter-arguments* (denoted **Arg**) can be used. Instead of the argument which refers to the evidence supporting the stated claim, counter-arguments demonstrate that the claim is not true. They can be used to derive metrics from counter-claims in GAM.

The inference rule is represented as a node of type *warrant* (denoted **W**). The warrant demonstrates in detail the argumentation strategy and justify why the inference rule used is valid. *Assumption* nodes are also possible but they are omitted in the description as they are not used in GAM.

Finally, the evidence can be of type: claim or *fact* (denoted **F**). Facts contain information which does not need additional justification (because it is obvious) or information whose validity is demonstrated in external documents. In contrary, claims must be demonstrated by other arguments. This way (by justifying claims) a trust case develops into a tree structure composed of many levels of abstraction. An example is given in Fig. 4.

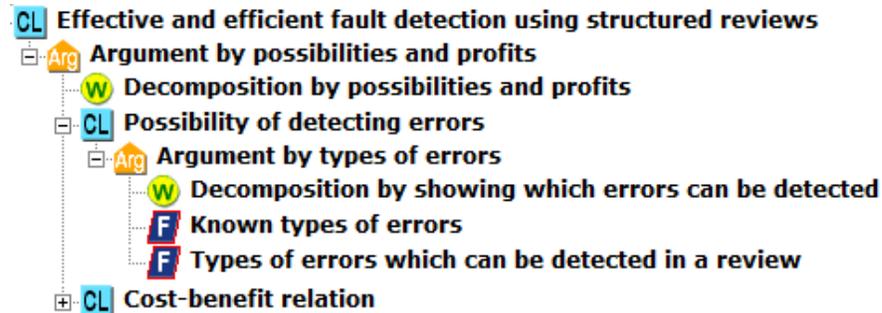


Fig. 4. Trust case example. Demonstrating structured reviews effectiveness and efficiency by showing that it is possible to detect errors of different types and that the benefits outstrip the cost.

Facts which are based on information contained in external documents can be supported by a node of type reference (denoted **Ref**). Such nodes contain information about the location of documents (usually it is a URL).

Additionally, anywhere in the argument tree an *information* node (denoted **i**) can be placed. Such nodes contain explanatory information which does not constitute a part of the proper argument.

Each of the above-mentioned nodes can be represented as a *link* to specific part of the trust case (if this part is to be re-used). Depending on where the link points at, it is represented by **CL**, **Arg**, **W**, **F**, **Ref** or **i**.

6 Case Study: Overview of the problem

In the case study we aimed at deriving metrics and direct measurements for the assessment of the effectiveness of *Standards Conformity Framework* (SCF) [2, 7].

SCF itself is part of the (broader) Trust-IT framework. SCF supports application of standards at the stages of achieving, assessing and maintaining compliance. The framework provides mechanisms which help to gather the evidence and present it in a legible way. The central component of SCF is a *Trust Case template* - a data structure derived from a given standard. Templates also include extra-standard data sources like guides, historical data, experts' knowledge, results of standards analyses and so on. All this information is kept in one electronic document. Such documents can be further assessed by auditors and if accepted, can be reused in many standards' compliance projects. SCF is supported by an on-line tool which enables teamwork while producing, gathering, and structuring the evidence which demonstrates the compliance with a standard.

SCF has already been used in some projects and the results are promising. To provide for a more objective assessment, a research program was initiated targeted at better understanding the benefits resulting from the framework application. As the overall goal, the analysis of the SCF's effectiveness was selected.

The objective was to derive metrics and direct measurements which could then be used in experiments to gather the necessary data needed to assess and demonstrate the effectiveness of SCF. The identified scope of direct measurements is going to be used while planning for a series of experiments targeting at the assessment of the effectiveness of the SCF framework.

Initially, we applied GQM to identify the scope of data to be gathered and the scope of metrics to be constructed from those data. The results were, however, not satisfactory although we had run two iterations of the GQM process to find the appropriate set of metrics. The major problems encountered were related to the derivation tree complexity and the scope management.

The complexity of the GQM tree resulted from the complexity of the problem itself (objective reason) and from the difficulties in defining the scope of data to be gathered (subjective reason). The scope of possible questions to be considered and possible metrics to provide answers to those questions was particularly broad also because we had to consider different variants. Therefore, deciding if a given question is beyond the scope or if the whole set of questions is complete was particularly difficult. In addition, while planning for the data gathering experiments it was often difficult to assess how a given data item influences the result of the measurement program.

The above difficulties led to the decision of applying argument structures to better control the relationship between the measurement objective, the metrics and data collection. Trust cases and the TCT supporting tool were chosen as the way to represent and maintain the argument structures.

7 Case Study: Application of GAM

To support derivation of metrics and measurements we created a trust case template of appropriate structure (see Fig. 5). It represents the whole measurement plan and is composed of four branches:

- (1) *'Effective support for achieving and assessing the compliance'* is the topmost claim (representing the measurement goal) which contains the whole argument structure. This claim is to be supported by the argument which justifies it (not shown in Fig.5).
- (2) *'Explanation'* contains additional information like the definitions of terms used to describe metrics and measurements.
- (3) *'Metrics Directory'* is the list of all metrics derived from the measurement goal.
- (4) *'Direct Measurements Directory'* contains the list of all direct measurements derived from the metrics.

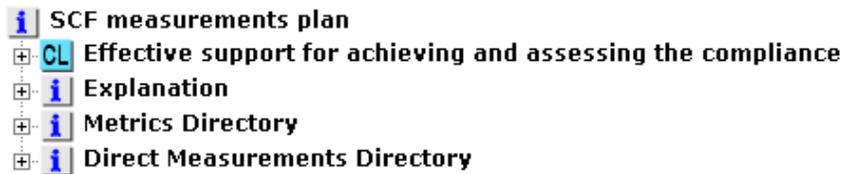


Fig. 5. SCF measurement plan trust case - a tree composed of the argument structure, explanations, a list of metrics and a list of direct measurements.

In the next step, the argument structure was developed. The measurement goal was decomposed into three claims and a warrant which describes the inference rule used. This is presented in Fig. 6.



Fig. 6. First level of decomposition. Arguing SCF effectiveness by demonstrating possibility of developing sound TC templates which positively influence the process of achieving the compliance and increase efficiency of the assessment.

In Fig. 6, it is argued that *'Effective support for achieving and assessing the compliance'* is provided because it is possible to create sound templates (represented by the claim *'TC templates development'*), application of the templates positively

influences the resulting level of compliance (represented by the claim ‘Achieving the compliance’) and the performance of assessing the compliance is significantly improved (represented by the claim ‘Assessing the compliance’).

The decomposition of the argument is justified by the ‘Decomposition into SCF application stages’ warrant which is further refined in Fig. 7.



Fig. 7. Argument supporting a warrant. Arguing that SCF effectiveness can be demonstrated by demonstrating three claims related to: development of templates, application of SCF at the stage of achieving the compliance, and application of SCF at the stage of assessing the compliance by the detailed analysis of the SCF application process.

The lower warrant (shown in Fig. 7) refers to the SCF application process structure and recalls the process diagram (through the link ‘SCF application process diagram’). The analysis of this process (included in the body of the warrant ‘Descriptive analysis of SCF application process’) explains why the structure shown in Fig. 6 is sufficient to assess effectiveness of SCF.

In the same way the three claims represented in Fig. 6 were decomposed into more refined claims and justified by more refined arguments. Each time, appropriate warrants were provided constraining the scope and giving the reason for decisions.

Finally, at a certain level of abstraction, to justify the higher-level claim it was enough to directly refer to measurable properties. At that level, the decomposition process stops. This last step is illustrated in Fig. 8.

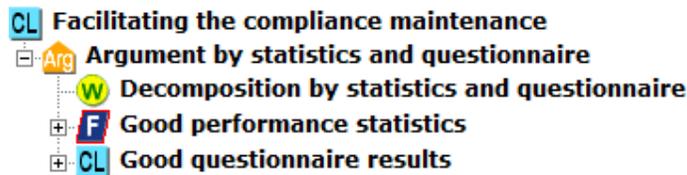


Fig. 8. Introduction of assertions in the argument structure. Demonstrating that compliance maintenance is facilitated by SCF because the statistics show improvement in the performance, and subjective opinions stated in questionnaires were positive.

Each leaf of the argument structure refers to a metric. The metric represents a measurable value having a certain business meaning, which can be an aggregation of a few measurements. In this way the method supports definition of the most suitable metrics. Additionally, the leaves contain assertions which impose constraints on values of the metrics. An assertion states that a given metric m is in a certain subset of possible values A as shown in (1).

$$M \in A; \text{ where} \tag{1}$$

M is a metric
A is a subset of possible values of m

For instance, in Fig. 8 the claim ‘*Facilitating the compliance maintenance*’ postulating that SCF facilitates the compliance maintenance is argued using performance statistics and questionnaire results. The claim ‘*Good questionnaire results*’ is decomposed further giving assessment criteria for answers to particular questions (explaining what ‘good’ means in this context). The fact ‘*Good performance statistics*’ is directly connected with a metric. It states that dealing with a change takes less than 90% of time needed if SCF were not used.

Fig. 8. also shows that at the same abstraction level it is possible to have claims and facts simultaneously. This gives flexibility in structuring the argumentation tree according to the needs.

Finally, all the claims must be refined into assertions. The number of levels of abstraction is dictated by the problem itself. At the bottom of the argument structure we will find claims which are supported by assertions only. (See Fig. 9)

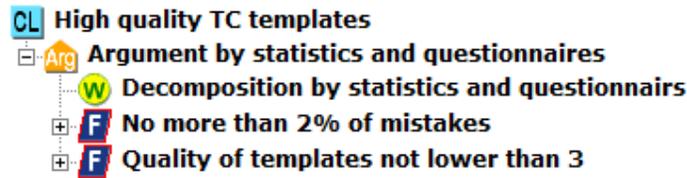


Fig. 9. Claim demonstrated by assertions only. Arguing high quality of TC templates by showing the statistics about mistakes and presenting results of questionnaires.

In the example above, the claim ‘*High quality TC templates*’ demonstrates that the templates developed according to the procedures defined by SCF are of high quality. It is justified by the requirement that the number of mistakes reported relates to less than 2% of requirements of the standard (represented as fact ‘*No more than 2% of mistakes*’) and the result of questionnaires used to assess the quality of templates generated with the help of SCF is at least 3 in the (1,..,5) scale (this is represented by fact ‘*Quality of templates not lower than 3*’).

In the next step the assertions were used to derive metrics. All the identified metrics were collected as facts in the ‘*Metrics Directory*’ branch shown in Fig. 10.

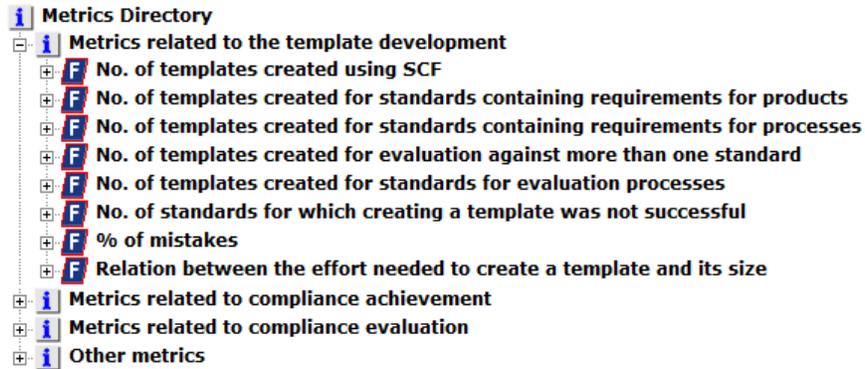


Fig. 10. Metrics directory.

To provide for traceability in both directions (i.e. from assertions to metrics and from metrics to assertions) under every assertion an information node is added which contains a link to the metric used by the assertion (as in Fig. 11).

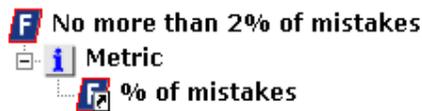


Fig. 11. Binding assertions and metrics. An assertion and a link to the metric derived from the assertion.

Let us consider the assertion shown in Fig. 11. It refers to a metric representing the per cent of mistakes in descriptions of requirements contained in templates. To construct such a metric we need raw data (a direct measurement). In general, a given metric M can be treated as a function $f(\)$ of several direct measurements dm_i as described in (2).

$$M = f(dm_i); \text{ where} \tag{2}$$

dm_i is a direct measurement
 f is a function

For instance, Fig. 12 gives an example metric and the related direct measurements.

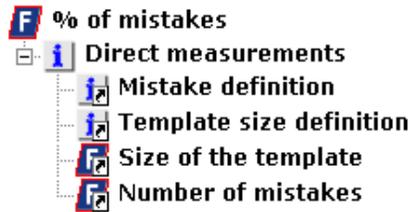


Fig. 12. Binding metrics and measurements. A metric, links to the direct measurements derived from the assertion and definitions of the notions needed to precisely express the metric.

The metric representing the number of mistakes in a template can be obtained using two direct measurements: one assessing the size of a template (represented as a link to fact ‘*Size of the template*’) and another one, assessing the number of mistakes in a template (represented as a link to fact ‘*Number of mistakes*’). Additionally, two links to information nodes, which contain the definitions of the notions used (the definitions of ‘mistake’ and ‘template size’) were added. The list of all the direct measurements is located in the branch ‘*Direct Measurements Directory*’ of Fig. 5.

8 Summary

In the paper the GAM method of systematic derivation of metrics and measurements from measurement goals was presented. The method was compared with GQM, one of the most popular methods of this type. In addition, a case study of application of GAM was described in detail, showing its most crucial aspects.

Application of GAM led to satisfactory results and removed the difficulties we have faced while applying GQM. The method proved to be more effective while solving this particular problem. The initial investment in development of GQM tree took about 24 hours in each of the two iterations. By contrast, application of GAM required only 10 hours¹.

The authors are fully aware that a single case study is not enough to draw more general conclusions related to comparison of the two methods. However, the results obtained are very encouraging and GAM is ready to use together with its supporting tool. We are planning for more case studies to provide more evidence on effectiveness of the method.

The results presented in this paper have been achieved in the context of the broader research program related to application of argument structures in various contexts. Except measurement plans, trust cases have been already applied to argue safety, security and privacy of e-health services and to support application of security standards. The method is supported by a matured, ready to use tool, which has already been used in a few projects e.g. EU 6th Framework Integrated Project PIPS and EU 6th

¹ It is worth mentioning however, that the GQM analysis was performed by a person without much prior experience with the method, and the GAM method was applied by a person having already some experience with the Trust-IT framework.

Framework STREP ANGEL. The tool provides effective means of editing the argumentation trees diminishing the difficulties related to maintenance, complexity and change management.

References

1. Briand L. C., Differing C., Rombach H. D., Practical Guidelines for Measurement-Based Process Improvement, Software Process. Improvement and Practice, No. 4 (1996)
2. Cyra Ł., Górski J.: Supporting compliance with safety standards by trust case templates, In Proceedings of ESREL 2007 (2007)
3. Górski J. et al.: Trust case: justifying trust in IT solution, Reliability Engineering and System Safety, Volume 89, Elsevier, Pages 33-47 (2005)
4. Górski, J.: Trust Case – a case for trustworthiness of IT infrastructures, In Kowalik J. & Gorski J. & Sachenko A. (Eds.), Cyberspace Security and Defense: Research Issues, NATO ARW Series, Pages 125-142, Springer-Verlag (2005)
5. Górski J.: Collaborative approach to trustworthiness of infrastructures, Proceedings of IEEE International Conference of Technologies for Homeland Security and Safety, TEHOSS 2005, Pages 137-142 (2005)
6. Górski J., Trust-IT – a framework for trust cases, Workshop on Assurance Cases for Security - The Metrics Challenge, DSN 2007 The 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, June 25 - June 28, Edinburgh, UK (2007)
7. IAG, Information Assurance Group Homepage, <http://iag.pg.gda.pl/iag/>
8. Lavazza L.: Providing Automated Support for the GQM Measurement Process, IEEE Software, Volume 17, Issue 3, Pages 56-62 (2000)
9. van Solingen R., Berghout E.: The Goal/Question/Metric Method: a Practical Guide for Quality Improvement of Software Development, Cambridge University Press (1999)
10. TCT User Manual, Information Assurance Group, Gdansk University of Technology (2006)