Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

# An Extended Goal-oriented Development Methodology with Contextual Dependability Analysis

Danilo F. Mendonça

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Orientadora
Prof. Dr.ª Genaína Nunes Rodrigues

Brasília
2015

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Mestrado em Informática

Coordenadora: Prof.ª Dr.ª Alba Cristina Magalhaes Alves de Melo

Banca examinadora composta por:

Prof. Dr.ª Genaína Nunes Rodrigues (Orientadora) — CIC/UnB
Prof.ª Dr.ª Vander Alves — CIC/UnB
Prof. Dr. Luciano Baresi — Politecnico di Milano

**CIP — Catalogação Internacional na Publicação**

Mendonça, Danilo F..

An Extended Goal-oriented Development Methodology with Contextual Dependability Analysis / Danilo F. Mendonça. Brasília : UnB, 2015.
26 p. : il. ; 29,5 cm.

Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2015.

1. LaTeX, 2. metodologia científica

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

# An Extended Goal-oriented Development Methodology with Contextual Dependability Analysis

Danilo F. Mendonça

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Prof. Dr.ª Genaína Nunes Rodrigues (Orientadora)
CIC/UnB

Prof.ª Dr.ª Vander Alves
CIC/UnB

Prof. Dr. Luciano Baresi
Politecnico di Milano

Prof.ª Dr.ª Alba Cristina Magalhaes Alves de Melo
Coordenadora do Mestrado em Informática

Brasília, 30 de janeiro de 2015

# Dedicatória

# Agradecimentos

# Resumo

A static and stable operation environment is not a reality for many systems nowadays. Context variations impose many threats to systems safety, including the activation of context specific failures. Goal-oriented software-development methodologies (SDM) adds the ‘why’ to system requirements, i.e., the intentionality behind system goals and the means to meet then. Contexts may affect what requirements are needed, which alternatives are available and the quality of these alternatives, including dependability attributes. In order to allow a formal and probabilistic analysis of systems affected by context variation and elicited with Goal-Oriented Requirements Engineering (GORE) approach, we have proposed an extension to the TROPOS goal-oriented methodology to include dependability constraints to goals and to provide a more precise and formal requirements verification by translating a contextual goal-model annotated with a behavioural regular expression into a PRISM probabilistic model to be checked against properties defined with the Probabilistic Computation Tree Logic (PCTL). We evaluated the proposal with a case study of a Mobile Personal Emergency Response System (MPERS).

**Palavras-chave:** LaTeX, metodologia científica

# Abstract

**Keywords:** LaTeX, scientific method

# Sumário

# Lista de Figuras

# Lista de Tabelas

# Capítulo 1

# Introduction

## 1.1 Problem definition

According to Lamsweerde, a poor requirements engineering (RE) is the major source of system failures. Lack of user involvement, requirements incompleteness, changing requirements, unrealistic expectations and unclear objectives are common causes [AXEL]. Goal-oriented requirements engineering (GORE) has gained the attention of both academic and industrial practitioners due to its ability to systematically model the intentionality behind system requirements. More than just presenting the ‘what’ and the ‘how’, goal models also express the ‘why’ of different requirements to exist. Also, its simple graphical notation allows non-technical stakeholders to take part in the analysis process and have a clear view of the system-to-be. Finally, automated model verification should avoid inconsistencies in the requirements specification.

In traditional GORE methodologies [GORE CA comparison], contribution analysis are based on domain knowledge about the positive, neutral (implicit) or negative impact of a given system alternative to one or more system goals, generally a qualitative goal. By comparing the overall contribution of two or more alternatives, a decision is made about which one should be adopted for the system-to-be. For instance, if one goal is to communicate with a remote user mobile, alternative means for the notification agent could be to send a SMS, an internet based message or a voice call. These alternatives may contribute with different values for qualitative goals such as ‘reliable delivery’, ‘fast delivery’, ‘convenient delivery’, etc.

The problem with this approach is threefold. First, it is based on domain knowledge information that may not exist or may not be precise and reliable. As a consequence, the decision of which alternative to use, either at design time or at runtime, may be biased and lead to unexpected violations - the selected alternative was actually unable to fulfil its qualitative goals. Second, it is limited to a static representation without any dynamic information that could be used to verify more complex attributes depending on the bigger picture of the system behaviour and its many nuances such as optional, alternative and interleaved executions. Finally, contribution links are deterministic. As such, the design decision based on contribution analysis is reduced to a simple sum comparison of the concurring alternatives with no support for probabilistic system properties.

Additionally to the contribution analysis limitation, the context in which systems operate may not be static. Mobile and pervasive computing, among others, are examples of new computer paradigms for which the environment is not static, but dynamic. Battery, signals strength, component's availability and the quality of physical resources and relevant information such as the user geographic location may vary through time, posing a new sort of challenge to the development of socio-technical systems based on these paradigms. The contextualization of the informations gathered at RE phase becomes imperative once its validity may be threatened by changing environment conditions [Finkelstein, CGM]. Moreover, any improvement for the GORE contribution analysis should also address multiple operational contexts.

## 1.2 Proposed solution

In order to provide a more solid and precise approach for the non-functional verification of different system alternatives and to improve the GORE contribution analysis, we propose the extension of the TROPOS goal-oriented software development methodology with a probabilistic model checking (PMC) approach that has already been explored and is supported by tools such as PRISM model checker[genaína PMC, PRISM]. The resulting verification model should represent activities that fulfils the root goal (global) or any lower level goal (local) and its elicited alternatives. The model should be checked for properties in Probabilistic Computation Tree Logic (PCTL) language and will provide estimations for attributes related to non-functional requirements such as the its local power consumption or its global reliability.

The PMC technique requires a probabilistic model representing system activities. As the goal model proposed in TROPOS is static, no information regarding achievement/execution order, cardinality and priority of goals/tasks is available, except the activity diagram for the detailing of an agent's single capability behaviour and the sequence diagram for agents interaction. Nonetheless, this problem was tackled by Dalpiaz et al. with the use of a regular expression language associated to goals, tasks and dependencies in a goal model to express information such as how many times the same goal should be achieved and the parallel or sequential tasks execution order [RGM]. We have used this proposal to fill the gap between the static goal model and its dynamic representation.

Finally, this dynamic view of the goal model may be translated to a probabilistic model following PMC technique. Our work uses the PRISM model checker and its language for this purpose.

To address a dynamic context of operation, context variables and its effects over goals, means and metrics should be parametrized to produce a formula that can check the system and its alternatives for different contexts of operation. This verification, performed as part of the Validation & Verification (VV) phase in RE, should anticipate any violations to the non-functional constraints. Treating a detected violation at design time may correspond to actions such as making a different choice for underlying components used by this alternative's tasks, optimization of its behaviour specification or even the disposal of this alternative as a means to satisfy its goal if there is at least one valid alternative. PMC technique also allows the identification of system alternatives with more influence on each metric through sensitive analysis.

Runtime self-adaptation is not covered by the scope of this work. However, based on the contextual analysis provided by the CGM and the enriched analysis provided by the verification of different alternatives using the PMC technique, it should not be difficult to extend the approach with the additional monitoring, planing and execution capabilities of a self-adaptation loop and have a self-adaptive architecture and mechanism reflected upon its runtime goal model requirements. This runtime facet of this proposal should be addressed in future work.

## 1.3 Evaluation

This proposal was evaluated with the application of the extended TROPOS methodology to the development of a Mobile Personal Emergency Response System (MPERS). This system may be seen as a body area network (BAN) with extended functionalities related to emergency response and mobile computing [BAD]. Instead of a home or hospital static environment, the MPERS is conceived to allow patients with different health risk degrees to maintain mobility while they are monitored and assisted. If a medical emergency is detected, a geolocation feature should identify where the emergency response team must be addressed to. The MPERS features were based on real emergency response systems available at the industry and also at the BAN proposed by Fernandes[Fernandes].

The evaluation process has pointed out the major benefits and limitations of the extended TROPOS proposal. Time to market and complexity is an important aspect for any software development methodology, therefore an automated generation of the PRISM probabilistic model was implemented based on an existing open source tool for TROPOS named TAOM4E[citation] in order to optimize the verification step by abstracting the

PRISM language from the analysts and reducing the effort to build the PRISM model. Also, the soundness and precision of the proposed probabilistic verification is crucial and must be evaluated as they should not result in mislead decisions about which alternatives should be used by the system. Instead, they must eliminate any violation that could lead to a system failure, specially severe or catastrophic failures.

## 1.4 Summary of Contributions

This section summarizes the contributions of this proposal.

1. A new contribution analysis approach for the TROPOS Goal-oriented software development methodology.
2. Conversion rules among different decomposition and runtime constraints in a runtime goal model to a PRISM probabilistic model.
3. Inclusion of context effects over goals, means and metrics in the PRISM model using appropriate constructs and parameters for each case.
4. A parser implementation for the regular expression (regex) language used in runtime goal-models to specify temporality, cardinality and goals priority.
5. An automatic generation of the PRISM model representing activities from a TROPOS goal-model annotated with the runtime regex and graphically modelled using the TAOM4E tool that supports TROPOS methodology.

## 1.5 Document organization

This dissertation is organized as follows. Chapter 2 presents the base concepts of this work and the most important related works. Chapter **??** details the problem tackled by this proposal. Chapter 4 presents the new extended TROPOS methodology, the rules for the translation between the contextual goal model and the probabilistic verification model, the parser for the runtime regex and finally the implementation approach for the automatic generation of the probabilistic model in PRISM language. Chapter **??** evaluates the proposal and describes its benefits and limitations. Finally, Chapter **??** concludes this work with final considerations about the current proposal, related proposals and our future work.

# Capítulo 2

# Baseline

## 2.1 Goal-oriented Requirements Engineering

Goal-oriented requirements engineering brings forward the intentionality behind system requirements. More than just presenting the *what* and the *how* of a system-to-be, it provides the justification for each requirement, that is, they also present the *why*. Through a directed graph tree that begins with a root goal, goals are connected trough decomposition links. Root and higher level goals are related to strategical concerns, while lower level and leaf-goals are related to technical and operational features of the system.

The main purpose of a goal model is to support the early process of RE, including the elicitation of social needs and dependencies, the actors involved in delivering functionality and resources, the decomposition of higher-level goals into more granular and detailed requirements chunks, the operationalization through means-end tasks and finally the comparison between different alternatives for the system-to-be. A goal model is said to be valid and complete if it follows all its syntactic rules and if all system goals are either decomposed, delegated to other actors or fulfilled by operational system tasks.

Three frameworks/methodologies, namely KAOS, i* and TROPOS, represent the foundations for the goal model analysis used by a variety of other proposals [KAOS, i*, TROPOS]. Despite some differences among their syntax, they all share a set of core concepts:

Entities

- **Actor:** an entity that has goals and can decide autonomously how to achieve them. They represent a physical, social or software agent. E.g.: A patient, an emergency center, a doctor and a Mobile Personal Emergency System running in patient's smartphone.

- **Goal:** are actors' strategic interests. A goal with a clear-cut criteria for its satisfaction is called a hard goal. In opposition, softgoals has no clear-cut criteria for deciding whether they are satisfied or not and are usually associated to non-functional requirements of an actor. E.g.: vital signs are monitored, emergency is detected, emergency center is notified (hard goals) and system availability, detection precision, emergency awareness (softgoals).

- **Task:** an operational means to satisfy actors' goals. E.g.: monitor temperature sensor, persist vital signs data, request emergency assistance.

Relations

- **AND/OR Decomposition:** a link that decomposes a goal/task into sub-goals/sub-tasks, meaning that all (at least one) decomposed goal(s)/task(s) must be satisfied/executed in order to satisfy its parent entity.

- **Means-end:** a means to fulfil an actor's goal through the execution of an operational task by the same actor.

- **Contribution link:** a positive or negative contribution between a given goal/task to a softgoal. Contribution links are used for deciding between alternative goals/tasks at design time (contribution analysis).

## 2.2 Contexts

Context may be defined as the reification of the environment that surrounds the system operation [FINKElSTEIN]. Contexts, as already stated, may not be static, but dynamic. An actor, that may be a system, has no control over its context of operation. Accordingly, an actor must be able to support different contexts of operation without violating its goals. Moreover, systems should be able to monitor the state of its surrounding environment and decide which alternative means will be used to fulfil its goals, as some may only be valid or optimized in specific contexts.

In GORE, dynamic contexts may affect what goals a system have to reach, the means available to meet them and also the quality achieved by each alternative[CGM]. Root goal and higher level strategical goals are not contextualized as they represent the main purpose of a system [Finkelstein]. As these goals are decomposed in more granular sub-goals, a context condition may dictate:

- If the goal is required for that context, limiting 'what' a system should do;
- If a sub-goal or task is adoptable, limiting the 'means' to fulfil a required goal;
- The positive, neutral or negative contribution of using some goal or task to another goal, usually a qualitative softgoal;

This last effect is the main focus of this work, as it is related to the GORE contribution analysis that we aim to improve.

## 2.3 Dependability Analysis

The concept of dependability is related to dependence and trust as well as the ability of a system to avoid failures that are more frequent and more severe than certain threshold. According to Avizienis et al., dependability encompasses the following attributes [AVIZIENIS]:

- Availability: readiness for correct service.
- Reliability: continuity of correct service.
- Integrity: absence of improper system alterations.

- Safety: absence of catastrophic consequences on the user(s) and the environment.

- Maintainability: ability to undergo modifications and repairs.

Correctness is opposed to failure. A failure is a perceived deviation from system expected behaviour that may have variable degree of consequences on the user(s) and the environment. Failures are caused by a specification fault(s) or specification violation(s). In the first case, either the goals or the means to fulfil then are incorrect or incomplete. In the second case, system implementation did not followed its operational specification due to a faulty implementation or a deviation from normal behaviour took place in one or more components involved in the execution.

The scope of this work is restricted to specification violations. That is, we assume that a system specification is valid and has no false assumptions, incompleteness or inconsistencies. Failures are caused by anomalies in the technical components or social actors participating in the execution of system tasks. Dependability analysis is used to provide estimations about different attributes related to system failures. These metrics may be specified as non-functional requirements for isolated system functionalities or for the whole system. Instead of softgoals, we use meta-requirements over functional goals with clear-cut quantitative criteria such as ‘99.999%’ reliable - a probabilistic value to make it compatible with the PMC estimation results.

## 2.4 PRISM Probabilistic Model Checker

The state based, probabilistic model checking technique used in this approach is supported by the PRISM model checker tool [PRISM]. PRISM allows the modelling and analysis of systems which exhibit random or probabilistic behaviour. The decision of using PRISM as the probabilistic state based model checker was due to the number of successful case studies that have used this tool, indicating its maturity [PRISM CS], and also due to its rich environment that is able to represent different kinds of probabilistic models and their evaluations.

A model checking is a formal method that may be used for quantitative and qualitative analysis of dependability attributes as long as:

- A formal system model may be built;

- Properties representing dependability attributes may be defined;

- The analysis overhead is justified, e.g., by its criticality.

As it will be explained in later sections, goal models may be easily extended with the proper information required for the verification of some important dependability attributes. The objective is to anticipate non-functional dependability violations and to support the decision of which alternatives to use in the system-to-be.

PRISM may be used for many different kinds of model evaluations depending on the abstraction level, the type of probabilistic model and the PCTL properties to be analysed. PRISM language offers a rich set of constructs that may represent system modules and components, among others architectural and design configurations.

It will be up to the analyst and stakeholders to define which type of probabilistic model and which PCTL properties must be analysed for each different system. Dependability attributes may be relevant for any sort of system, but are certainly important for systems with some criticality degree, i.e., for those whose failure could have severe or catastrophic consequences for the user(s) and for the environment.

# Capítulo 3

# Related Work

## 3.1 Contextual Goal Model

The Contextual Goal Model (CGM) [CGM] proposed the contextualization of required goals, adoptable means (goals/tasks) and contribution links values. The main benefit of this work is to enrich the original goal model with the contextualization of entities and relations affected by context variations and to provide a rationale for context analysis. In contrast, the main problem tackled by the current work is the verification of non-functional attributes that requires a more precise and formal approach instead of the existing contribution analysis that is based on analysts direct evaluation of the forward impact between goals/tasks and softgoals.

In this regard, the CGM provided more realistic and precise contribution analysis contextualized by environment conditions, but did not change the nature of the contribution analysis process. Our work has benefited from the CGM conceptual model and has extended the non-functional GORE analysis with a context-dependent formal verification, i.e., that includes different context effects in the probabilistic model used by the PMC to estimate the values of required non-functional attributes of the system and provide a reliable decision criteria for the selection of concurring alternatives in the goal model before it is implemented.

## 3.2 Awareness Requirements

Souza et al. [AwaReq] proposed the Awareness Requirements (AwReq) as a meta-requirement in a goal model, i.e., AwReq specify the success/failure rate and temporal constraints for other requirements in the model, including goals, softgoals, tasks and other AwReqs (*-meta-requirement). The purpose is to enrich the original goal model and provide clear-cut criteria for self-adaptation, as runtime AwReq violations should be addressed by corrective actions. AwReq are formalized by a temporal logic formula, namely the Object Constraints Logic with Temporal Message (OCLtm).

Despite its contribution to the specification of meta-requirements in the goal models, AwReq do not provide an approach to analyse and validate its meta-requirements before system implementation and monitoring. Original GORE contribution analysis could be used to define the impact of a given alternative to some attribute or value composing the AwReq. However, the paper focuses only on attributes that can be monitored by the system at runtime. In contrast, our approach relies on the improved contribution analysis, i.e., the model based verification of attributes through PMC technique that can be performed at design time and provide alternative design decision criteria. Moreover, a similar meta-requirement is used by our approach to define PCTL properties that must be checked by the PMC. These properties are also associated to system goals.

## 3.3 Dependability Contextual Goal Model

This work has been preceded by another proposal concerning goal-oriented requirements engineering, dependability analysis and dynamic contexts, namely the Dependability Contextual Goal Model (DCGM) [DCGM]. The contribution was focused on both dependability requirements and estimations based on declarative rules and a variable context of operation.

In DCGM, a failure classification scheme was used to classify the consequence level and domain of failures in achieving system goals. This process lead to the definition of dependability constraints that must be achieved by the means-end tasks used to fulfil leaf-goals in specific contexts of operation, i.e., to the specification of contextual dependability requirements. These requirements inherited the same concept of the AwReq, but instead of being static, they could be associated to a context condition. Another facet of the DCGM is the contextual failure implication, which consisted of a dependability specific GORE contribution analysis supported by Fuzzy Logic to define IF-THEN rules between context conditions and the level of a dependability attribute, e.g., availability and reliability.

The main drawback of this proposal was the lack of scalability, as declarative rules must be provided for different goals, attributes and contexts, proving to be a time consuming task for the analysts. A second problem was the subjectivity of the rules, as they were based in domain knowledge. This problem, as much as in GORE contribution analysis, lead to the idea of coupling a more precise and reliable verification of non-functional requirements of a goal model based on the PMC technique. Still, the idea of a failure classification and the specification of contextual dependability requirements were kept, with the difference that now other attributes besides dependability ones may be specified.

## 3.4 Runtime Goal Model

Despite the use of goal models to support the monitoring and adaptation functions at runtime, Dalpiaz et al. argued that these works are ‘using design artefacts for purposes they are not meant to, i.e., for reasoning about runtime system behaviour’. As such, they proposed a conceptual distinction between the static goal model, named Design Goal Models (DGM), and the Runtime

Goal Model (RGM) that extend DGM with ‘additional state, behavioural and historical information about the fulfilment of goals’ [RGM].

The main purpose of the RGM approach is to provide the proper specification of behaviour information among system goals. RGM defines a class model, while the Instance Goal Model (IGM) provides the instance model that must conform to its class specification. IGM are useful to have an instance representation of the RGM provided by the monitoring of the activities involved in fulfilling system goals. If the monitored IGM violates the RGM, then a corrective action would have to take place. Again, our work has benefited from the conceptual contribution, this time by using the runtime regex language to have a behaviour specification for system goals and use it to generate the probabilistic model. In contrast, our work does not cover instance and monitoring aspects and focuses on the &V phase of RE to anticipate any violation and for the selection of the most appropriate concurrent alternative elicited for the system.

## 3.5 Formal TROPOS

The idea behind the formalization of a goal model, as proposed by Formal TROPOS [FTROPOS], is to provide a verifiable specification of sufficient and/or necessary conditions to create and achieve intentional elements and dependencies in the model and invariants for each element. In addition to these conditions, new *prior-to* links describe the temporal order of intentional elements and cardinality constraints may be added to any link in the model. Finally, Formal TROPOS uses a first-order linear-time temporal logic as a specification language.

The nature of the verification of a goal model with Formal TROPOS specification is different from the PMC used by our work. Formal TROPOS aims to provide the information required for a consistency verification of the model. The verification is not only for the abstract TROPOS syntax, but also to domain specific information of how intentional elements are created and fulfilled in time. Once the model starts to have more elements and relations, its consistency checking becomes non-trivial, justifying the use of a formal specification that can be verified by a model checker tool.

In our work, PMC technique is used to build a probabilistic model representation of the goal model enriched by dynamic specification (RGM) and enable the verification of properties that depend on how activities in the model are organized in terms of time, cardinality and priority and how each activity contributes to the property being verified. For instance, if power consumption is to be checked, each activity has to be associated to a power consumption unit and the global consumption value is evaluated considering any non-determinism specified in the execution workflow. Thus, even if the Formal TROPOS language allows the specification of dynamic aspects of a goal model, it is tailored for a consistency checking and not for the verification of non-functional requirements of the model, for instance dependability requirements.

# Capítulo 4

# Proposal

In the PMC technique adopted by this proposal, a behavioural specification, usually provided by UML activity and sequence diagrams, are manually converted to a probabilistic model in PRISM language. As a goal model goes from strategical root goal to operational leaf-goals, and each leaf-goal describes a desired state reachable by either a delegation to other actor or by a operational task, then a behaviour specification as proposed by the RGM may be seen as an activity diagram and be used to generate a probabilistic model in PRISM language. This allows the model checking of the corresponding goal model as a set of activities for which temporal and other behaviour aspects are specified by the runtime regex of the RGM.

- Making a different choice for underlying components: In some cases the replacement of a technical component for another of the same class can improve the quality of how they achieve their goal. For instance,

- Behaviour optimization: The quality may also depend on the pattern used for the activities execution. The specification of a different pattern may eliminate the non-functional violation.

- Contextualizing the alternative: An alternative may only violate a NFR in specific contexts. In this case, different valid alternatives may be used according to the context of operation.

- Alternative disposal: If the alternative is in absolute violation or if its validity is restricted to contexts that have at least one other valid alternative, this branch can be eliminated from the model.

To evaluate the current proposal with the MPERS case study, we have used the a discrete-time Markov chain (DTMC) probabilistic model and focused on the verification of properties related to dependability, i.e., the reachability of the final success state of a set of goal model activities that represents:

- if the set is composed of the minimum set of activities that satisfies the root goal: its global reliability;
- if the set is composed of the minimum set of activities that satisfies any lower-level goal: its local reliability.