From Natural Language to Formal Proof Goal

Structured Goal Formalisation Applied to Medical Guidelines (Extended abstract)

Ruud Stegers[1], Annette ten Teije[1], and Frank van Harmelen[1]

Vrije Universiteit, Amsterdam

Abstract. The main problem encountered when starting verification of goals for some formal system, is the ambiguity of those goals when they are specified in natural language. To verify goals given in natural language, a translation of those goals to the formalism of the verification tool is required. The main concern is to assure equivalence of the final translation and the original. A structured method is required to assure equivalence in every case.

This article proposes a goal formalisation method in five steps, in which the domain expert is involved in such a way that the correctness of the result can be assured. The contribution of this article is a conceptual goal model, a formal expression language for this model, and a structured method which transforms any input goal to a fully formalised goal in the required target formalism. The proposed formalisation method guarantees essential properties like *correctness*, *traceability*, *reduced variability* and *reusability*.

1 Introduction

The main problem encountered when starting verification of goals for some formal system, is the ambiguity of those goals when they are specified in natural language. No matter what domain, or what source of the goals: there are always many implicit assumptions and interpretations that must be made explicit before they can be used for formal verification. An ad-hoc method, in which the expert on the formal system makes the translation by hand directly into the logic of the target system, may work sometimes, but is error prone due to the obvious domain specific choices and interpretations that have to be made.

Incorporating a domain expert in the formalisation process seems to be a necessity, however the gap between the natural language representation and the logic of the verification tool is far to big to close without help. This article proposes a structured method, understandable by the domain expert and yet with enough expressive power for the formal methods expert. This article will focus on suitable representations and required steps for the formalisation of natural language goals. The contribution of this article is a common frame of reference

This work has been partially supported by the European Commission's IST program, under contract number IST-FP6-508794 Protocure-II.

for all the experts involved (the goal model), a formal expression language for this model (*GDL*), and a five-step method which transforms any input goal to a fully formalised goal in the required target formalism. The proposed formalisation method may be applied to any domain and guarantees essential properties like *correctness*, *traceability*, *reduced variability* and *reusability*.

Although the proposed method is domain independent, its origins can be found in the medical domain. An example from this domain will be used to illustrate the individual steps throughout this article. The goal shown below is used to verify medical guidelines as used by care providers. Those guidelines provide directives and instructions for the diagnosis and treatment of selected deceases and injuries. By applying verification techniques to guidelines, these may be improved: the aim is both to increase the quality of the care, and to prevent unnecessary medical tests and treatment.

Original - Example

> "The percentage of patients in the last year, with whom the possibility of breast reconstruction was discussed before mastectomy was performed."

The example goal is what is called an *indicator*: in hospitals indicators are used to measure the quality of the care on a periodic basis (typically each year). With help from a doctor and the proposed method, this indicator will be formalised so it becomes a suitable goal for formal verification.

The proposed formalisation method has been evaluated on several goals from the medical domain[1]. The four chosen goals apply to guidelines for treatment of diabetes, jaundice and breast-cancer. In the Protocure project[2, 3], these guidelines have been formalised for that purpose using Asbru, a plan oriented modelling language. In the same project, these models were translated to the temporal logic of KIV, a tool that allows the guidelines to be verified using symbolic execution[4]. The example of this article is one of those four goals. The resulting formalised goal has successfully been proven for the breast-cancer guideline using KIV.

In the next section the goal model which provides the shared frame of reference for both experts (e.g. medical expert, formal methods expert) is explained. Subsequently, Sect. 3 defines five requirements for the formalisation process followed by a discussion of the method itself. Section 4 discusses related work, followed by the conclusion in Sect. 5.

2 The Goal Model

To provide a common frame of reference for both the domain expert, and the formal methods expert, a high level goal model is required. This model is depicted in Fig. 1.

A goal is expressed in terms of a *start event*, a *(pre) condition* for this event, an *end event*, and some *desired behaviour* in between. From the moment the start event is seen while the condition is true, the process model (i.e. the guideline) should adhere to the prescribed behaviour for as long as the end event does not occur.

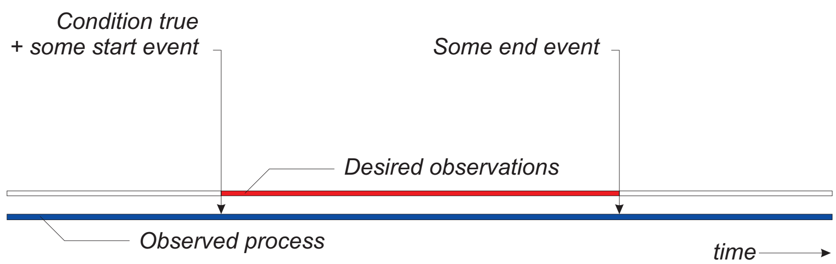

Fig. 1. Shared goal model

The simple nature of this shared goal model makes is easy to understand, however practical use has also shown that the model allows for sufficiently expressive goal descriptions. The next sections will show that the common goal model maps naturally to both natural language and to a formal representation (i.e. *GDL*).

3 The Formalisation Method

Having a shared frame of reference is only the start. The process itself of how to get from an arbitrary natural language goal to the target formalism via the goal model is equally important. The following requirements must be met by the formalisation process:

- **Ambiguities.** Identify and clarify all assumptions and ambiguities present in the original goal.
- **Correctness.** Ensure correctness of every change to the goal: the domain expert should be able to validate changes to ensure their validity.
- **Traceability.** Ensure traceability. The formalisation must be completely reproducible by means of the intermediate results and the documentation.
- **Reusability.** Enable reusability of work at different stages. Maintain generality for as long as possible.
- **Variability.** Reduce variability of the formalisation result.

The numbered steps depicted in Fig. 2 assure compliance with those requirements. The blocks on the left hand side represent the goal in natural language. On the right side the blocks represent the formal expression of the same goal. The individual steps will now be demonstrated for the example.

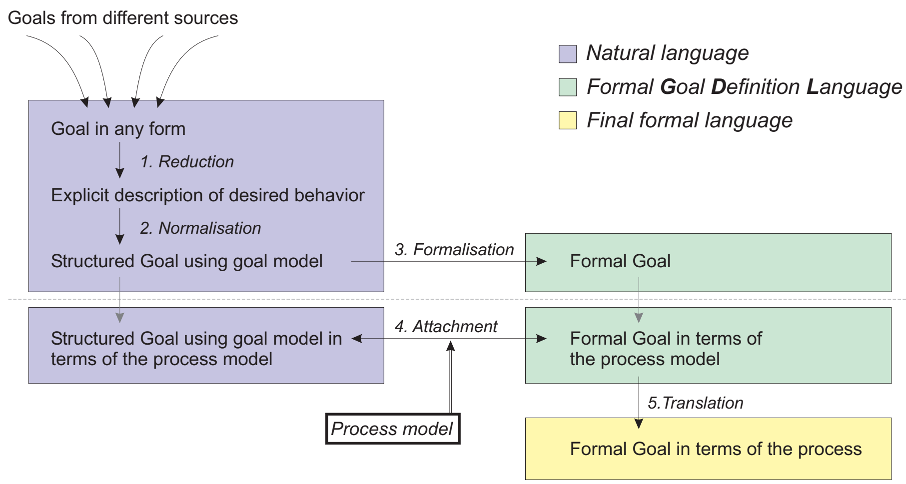

Fig. 2. Life-cycle of a goal

1. Reduction Due to the variety of the source of goals, the first step is to make sure that the desired behaviour is explicitly described. This explicit description preserves the quality aspects of the goal, while getting rid of non-essential information. The domain expert is the one primarily performing the reduction. The main responsibility of the domain expert in this and subsequent steps, is that the essence of the goal is not altered, or if it is, that this happens deliberate and well documented.

> **Reduction** - Example
>
> "The possibility of breast reconstruction should be discussed with all patients prior to mastectomy."

For the example (refer to Sect. 1 on page 2), the knowledge of the medical expert is used to establish that the percentage mentioned in the original version should preferably be as high a possible. This allows removal of references to this percentage (*the percentage of patients* becomes *all patients*). Additionally, by realising that this indicator is repetitively applied every year, it is concluded that in this case the patient must always be informed about possibilities for breast reconstruction prior to mastectomy:

2. Normalisation After the reduction, the next task is to rewrite the goal in terms of the goal model: some behaviour that should be adhered to between some start and some end event. Not only the terms, but also the structure of the goal model is imposed on the goal: the goal is transformed into a normal form.

The normal form consist of four elements in brackets (Condition, Start, End and the Behaviour), however, the whole sentence must be well-formed. During the normalisation ambiguities and implicit assumptions concerning temporal relations between events are almost automatically being taken care of: rewriting into this structure raises questions which, when answered by the domain expert, clarify the ambiguities of this kind. The reduction and normalisation are just another way of writing down the original goal. By not making adjustments specific to the object of verification (models), the normalised goal may be reused for verification of many models.

Normalization - Example

^{C}[For women with breast-cancer],^{S}[after start of the medical care] but ^{E}[before commencing mastectomy], ^{B}[the possibility of breast reconstruction should have been discussed with the patient].

3. Formalisation The next step is to transform the structured natural language version to a formalised version which can be used by the formal methods expert. The formal expression of the goal will be provided by *GDL*, the *Goal Definition Language*[1]. This newly developed language is specifically designed to reflect the structure of the goal model and will therefore also stay very close to the structured version. Due to this close relation, changes in one version are easily duplicated in the other version. This makes that the formal methods expert and the domain expert can discuss the same goal using their own representations.

The Goal definition Language, *GDL*, consists of two parts: A general part, *Generic GDL*, that represents the goal model itself, and a task specific extension to *GDL* that defines the exact conditions and events that can be used. The formalisation of the example is shown here in presentation syntax. Only *Generic GDL* is used at this stage for the formalisation:

Formalisation - Example

Goal Example
  Precondition
    For women with breast-cancer

  Time-specification
    From the start of the medical care
    Until start of mastectomy

  Observe-during-period ≥ 1
    discuss possibility of breast reconstruction with the patient

The elements from the goal model are easily recognised (Condition, Start, End and the Behaviour). Although the overall structure is fixed, several elements

[1] A full language specification including formal semantics are available in [1]

may be replaced. The most important of those is the behaviour. The available choice of elements raises questions which clarify behaviour related ambiguities. (e.g. the difference between 'observe once' or 'observe at least once'.)

4. The Attachment Given the formalisation result so far, the only thing left is to formalise the natural language parts with according to concepts available in the process model (i.e. the guideline model). The task consists of finding an equivalent concept in the process model, for every concept in the goal. During the attachment the domain expert makes sure that the concepts which are 'attached' are indeed equivalent, and where necessary, adjustments may be made. To be able to evaluate every adjustment, both versions of the goal (*GDL* and structured natural language) are kept synchronised: changes made to one, should be reflected in the other. That way, both experts can evaluate the (intermediate) result using their own representation. Also here, documentation is essential in order to be able to verify the formalisation result. Notice the additional information about the model to which the goal has been attached to between brackets in the example.

Attachment - Example (BC Ch1 23.11.2005)

Goal Example
Precondition
always-true

Time-specification
From Transition ch1-treatment **enter** active
Until Transition mastectomy-proper **enter** active

Observe-during-period
Planstate patient-information-reconstruction = completed

Since the process model may be described by any formal language – each with specific elements and features – a specialised *GDL* extension is needed for the specific modelling language used. The model verified using the example goal is the Dutch guideline for treatment of breast cancer which has been formalised in Asbru. Therefore, an Asbru specific extension – ***GDL***-Asbru – was developed, which consists of Asbru specific conditions and events. Asbru is a plan-specification language defined as part of the Asgaard/Asbru project[5].

5. Translation The translation of *GDL* to the logic of the verification tool should be a strictly mechanical step. This is essential since changes at this stage would be impossible to detect and validate by the domain expert. The mechanical nature of the translation makes it trivial from a process perspective. The only consideration that must be taken into account is that the translation must have the same semantics as *GDL*.

The guidelines under investigation are being verified in KIV, a tool which – amongst others – allows symbolic execution of parallel programs[4]. To this end, the Asbru models were translated to parallel programs in KIV. For the goals, an efficient modular translation of *Generic GDL* and ***GDL***-Asbru to KIV has been made. Using this translation, KIV was successfully used to close the proof on the example goal.

4 Related work

In the software engineering area the same problem is investigated, namely the transition from natural language to a formal representation, for instance [6–8]. [6] presents the approach of lightweight formal methods which starts with natural language and ends up with a semi-formal representation. Our proposal goes further and continues to the end (i.e. to a full formal representation).

In contrast with [6], [7, 8] mainly focus on the transition from the semi-formal to the formal representation. This means the first step from natural language to semi-formal is rather large. In our case this step is more fine grained. Furthermore in our case the both experts (domain, formal methods) have their own representation. Whereas in [7, 8] one representation is used. The final result with [7, 8] consists of the pseudo-code respectively the objects of the application. Our method aims to end up with the logic of the verification tool. Notice that in requirements engineering part of the effort goes into identifying the requirements of the user, whereas for the proposed method of this article, the requirements in natural language are considered given.

5 Conclusion

This article has proposed a method of formalising natural language goals in such a way that the domain expert is involved in every step that may change the meaning of the goal. The contribution of this article is a common vocabulary between the domain expert and the formal methods expert in the form of the goal model. The five step method adds a structured and controlled way of rewriting goals via this goal model to any target formalism. Five requirements have been formulated in section 3, which will be evaluated below.

- **Ambiguities** During the formalisation process, each step targets different kinds of ambiguities. The biggest reduction of ambiguity is achieved during the normalisation. To rethink a goal in terms of the goal model automatically raises questions. During the formalisation details of the behaviour are made specific by forcing a choice for a specific kind of behaviour. During the attachment, conceptual ambiguities are resolved by connecting the concepts in the goal to available concepts in the model: by evaluating their equivalence the exact meaning needs to be established. By following the steps, ambiguities are naturally solved.

- **Correctness** The main instrument to achieve correctness is the continuous involvement of the domain expert. An up-to-date natural language version of the goal is maintained throughout the process which allows the expert to focus on the meaning of the goal in a familiar form. The domain expert decides whether a proposed change is correct or not. By means of the formal *GDL* semantics the correctness of the translation step can be confirmed.
- **Traceability** The main tool to achieve traceability is by adding documentation to the intermediate result after each step. Additionally, the subdivision in steps with specific tasks reduces the amount of required documentation: in the context of the task, many transformations are straight forward and don't need to be explained. In that respect, traceability follows from the method.
- **Reusability** During the formalisation process, there are two distinct points where the intermediate result may be reused. First of all, the formalisation result may be reused for different process models. The attachment result may be translated to different tools. One example of reuse of an attached goal would be the translation both to KIV and to the SMV model checking environment.
- **Variability** The task oriented subdivision of the formalisation process causes the first reduction in variability. The fixed order of steps works towards a uniform result. Additionally, the canonical forms force the result into the right direction. Finally, the vocabulary of the goal model in general, and of *GDL* specific, do not allow many different ways to express a single goal. Every step tries to achieve convergence to the unique *GDL* expression. The mechanical nature of the translation enforces invariance in the last step. The only real source of variance in the result are differences in interpretation by the domain expert of the original goal. However, this cannot be avoided.

References

1. Stegers, R.: From natural language to formal proof goal: Structured goal formalisation applied to medical guidelines. Master's thesis, Free University Amsterdam, department of artificial intelligence (2006) Available from: `http://www.stegers.info/Ruud/MastersThesis.pdf`.
2. ten Teije, A., et al.: Improving medical protocols by formal methods. Artificial Intelligence In Medicine **36**(3) (2006) 193–209
3. Deliverable D4.2: Specification of guideline properties & indicators, Protocure Project-II, IST-FP6-508794 (2006) `http://www.protocure.org/`.
4. Balser, M.: Verifying Concurrent Systems with Symbolic Execution. PhD thesis, University of Augsburg, Augsburg (2005)
5. Shahar, Y., Miksch, S., Johnson, P.: The asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines. In: Artificial Intelligence in Medicine. Volume 14. (1998) 29–51
6. George, V., Vaughn, R.: Application of lightweight formal methods in requirement engineering. In: CrossTalk. The Journal of Defence Software Engineering. (2003)
7. Bryant, B.R.: Object-oriented natural language requirements specification, Canberra, Australia (2000)
8. Cooper, K., Ito, M.: Formalizing a structured natural language requirements specification notation. In: Twelfth Annual International Symposium of the International Council On Systems Engineering (INCOSE), Las Vegas, Nevada USA (2002)