Beyond Goal Representation: Checking Goal-Satisfaction by Temporal Reasoning with Business Processes

Choong-ho Yi[1] and Paul Johannesson[2]

[1] Department of Information Technology
Karlstad University S-651 88 Karlstad Sweden
e-mail: choong-ho.yi@kau.se
[2] Department of Computer and Systems Sciences
Stockholm University Electrum 230 S-164 40 Kista Sweden
e-mail: pajo@dsv.su.se

Abstract. Most formal approaches to goals proposed within information systems engineering, view goals from the requirements engineering perspective, i.e. for producing future software. Typically, these approaches begin with extracting goals from informal reality and end with representing them in some formal language, leaving the questions arising afterwards unanswered: How can we check whether goals are achieved or not in real business processes? If the goals are not satisfied, why and what to do? This paper presents a formal approach to *representing and reasoning with goals* using a first order many sorted temporal logic, where goals are expressed in terms of actions and static and temporal constraints; the above questions are answered by model theoretic formal reasoning with goals and business processes.

1 Introduction

This paper presents a formal approach to *representing and reasoning with business goals*, an essential aspect in a business enterprise. There are two main differences between previous formal approaches to goals, e.g. [3], [2] and [4], and ours. First, goals there have been viewed from the RE (requirements engineering) perspective, i.e. for producing *future* software, while we approach goals as guiding principles for business activities in an *existing* enterprise. Second, the previous ones mainly begin with extracting goals from informal reality and end with representing them in some formal language, where questions like the following are analysed: What are the goals of an enterprise? How are the actors trying to achieve the goals? How can a goal be decomposed into subgoals? How can a goal be formulated? On the other hand, we start where the others left off, i.e. from formal representation of goals, and proceed to reasoning with them, addressing "post-formulation" questions like the following: Are the goals achieved in the business activities of the enterprise? How can we check it? If goals are not satisfied, why and what to do? The reason we do not consider the topic of goal extraction is not because it is unimportant, but because it has been studied widely elsewhere and our point is to show reasoning with goals.

M. Jarke, A. Oberweis (Eds.): CAiSE'99, LNCS 1626, pp. 462-466, 1999.

Most formal approaches within ISE (information systems engineering) do not fit our purpose well, mainly because they are not intended for the same purpose. A natural way to formulate goals of an IS which are highly abstract by nature is, to describe (implement) them in terms of the concepts which are less abstract but reflect the fundamental aspects of the system. *Time*, *actions* and *states* are such ones, and so should be formalised first in an adequate way. However, 1) time is not considered in, e.g. Z, VDM or Statechart and it is not clear how time can be integrated into them. 2) While in reality actions have duration, i.e. take time, and different actions may have different length of duration, they are often considered without time, e.g. in OASIS, TROLL and ERAE. In other words all actions are understood to be instantaneous, and we cannot express that some business activities (e.g. a supply from USA to Sweden by ship) may take longer time (say, a month) than others (e.g. an order by telephone). 3) In some object-oriented approaches, e.g. TROLL and KAOS, state is defined w.r.t. an object, i.e. state of an object. However, it may cause difficulties when reasoning about several objects at the same time unless all the states of the objects involved are synchronised.

We are using a first order many sorted temporal logic whose semantics has been based on Features and Fluents [6], a framework within the area knowledge representation. The choice of many sorted logic has been motivated to deal with different kinds of concepts effectively, e.g. different object types (e.g. Agent, Product, Item) in a system, properties (e.g. item_of(I,P) meaning "I is an item of product P"), actions (e.g. order(Ag1,Ag2,N,P) meaning "Agent Ag1 orders with reference number N a product P from agent Ag2"), time points, etc, and relationships over them. For example, the predicate holds(Pr,T) over Property×Time states that the property Pr holds at time T, and the predicate occurs(A,T1,T2) over Action×Time×Time states that the action A occurs over the time period T1 (start time) and T2 (end time). Please refer to [7] for a detailed description of the logic.

2 Business Rules

In our approach goals are implemented in terms of *business rules* that prescribe what and how to do in different business situations. These rules are expressed as wffs in the logic. For example, the rule

$$\begin{aligned} &\mathsf{max\_delivery\_time}(7) \equiv \\ &\quad \mathsf{occurs}(\mathsf{order}(Ag1,Ag2,N,P),T1,T2) \rightarrow \\ &\quad \exists!I \exists!T3 \exists!T4\ (\mathsf{holds}(\mathsf{item\_of}(I,P),T3) \wedge \\ &\quad \mathsf{occurs}(\mathsf{supply}(Ag2,Ag1,N,P,I),T3,T4) \wedge T2 \leq T3 \wedge T4\text{-}T2 \leq 7) \end{aligned} \qquad (1)$$

states that whenever a customer Ag1 orders a product P over [T1,T2], after order is completed (T2≤T3), an item I of the product type is to be delivered exactly once within 7 days (T4-T2≤7), assuming that all time points are expressing dates. Above ∃! is a syntactical abbreviation denoting the quantifier "exactly one", and all free variables are ∀-quantified. The next rule, with all free variables ∀-quantified, states, suppliers Ag1 are expected to stop any more delivery to those customers Ag2 who have not paid for previous shipments within 30 days.

stop_supply_after(30) ≡ (**2**)
occurs(supply(Ag1,Ag2,N1,P1,I1),T1,T2) ∧
occurs(invoice(Ag1,Ag2,N1,P1,N2),T3,T4) ∧
¬holds(paid(Ag2,Ag1,N1,N2),T4+30) ∧ T5-T4 >30 →
¬∃ N3∃P2∃I2∃T6 occurs(supply(Ag1,Ag2,N3,P2,I2),T5,T6)

The agents pair in actions is motivated to describe communications between multiple agents and to trace who is capable of (responsible for) what. Such information may be essential to a business enterprise. The use of explicit time points makes it possible to express action duration and temporal order between actions. Business rules represented as such, i.e. capturing dynamic, static as well as temporal aspects, provide rich expressiveness and flexibility, and are suited for implementing goals.

3 Goals

This section presents a formal approach to answer the "post-formulation" questions raised in Section 1, where goals are viewed as guiding principles for business activities in an existing enterprise. First, we show how goals can be formulated using business rules.

3.1 Implementing Goals Using Business Rules

A characteristic of goals is that they are highly abstract, e.g. customer_satisfied ("the customers should be satisfied"). In order to make such a goal operational, we decompose it into subgoals at a lower level using AND/OR graphs, e.g. AND(prompt_delivery, purchase_on_credit) which means "the customers should get prompt delivery" and "the customers should be allowed to make a purchase on credit". Decomposition process continues until the subgoals can be implemented in terms of business rules described in Section 2. For example, the subgoal prompt_delivery can be implemented by the rule (**1**), while the rule (**2**) may be used for, say, the goal measure_against_unpaid, "measures should be taken against the customers who do not pay in time", etc.

3.2 Goals as Guiding Principles for Business Processes

In order to answer the "post-formulation" questions by formal reasoning with business processes, we describe a business process in the language as a set of i) predicates occurs(A,T1,T2) and ii) formulae describing constraints on the action occurrences. The process may exist already in the enterprise, or be proposed as a possible process in the future. An underlying principle is it should be possible to check any goals on any business processes that may take place in the enterprise. Then the main idea behind the reasoning can be summarised as follows: A business goal is satisfied in a business process if at least one *interpretation* I exists which assigns true to both the process and (the business rule implementing) the goal. (An interpretation

is a mathematical structure that assigns truth values to each wff in the language. In logic this way of drawing conclusions based on interpretations is called model theory. Please refer to [7] for details on the reasoning not presented here for space reason.) We sketch the idea in three steps with examples below.

1) Let G be an arbitrary goal and let Bp be an arbitrary business process in which we would like to check the goal. For example, consider the goal

$$G \equiv \text{measure\_against\_unpaid}$$

and the process

$$\begin{aligned} Bp \equiv \{\ & \text{occurs}(\text{order}(ag1, ag3, 1001, p1), 11, 13), \\ & \text{occurs}(\text{supply}(ag3, ag1, 1001, p1, i2), 15, 19), \\ & \text{occurs}(\text{invoice}(ag3, ag1, 1001, p1, 28), 20, 21), \\ & \text{occurs}(\text{order}(ag1, ag3, 1002, p2), 50, 52), \\ & \text{occurs}(\text{supply}(ag3, ag1, 1002, p2, i1), 53, 59), \\ & \exists I\ (\text{holds}(\text{owner\_of}(ag3, I), 15) \wedge \neg(I = i2))\ \} \end{aligned}$$

where ag3 supplies and invoices to ag1 w.r.t. the order #1001; but ag1 places another order #1002 without paying for the previous order and ag3 supplies again; ag3 is to have at least one more item than i2 when he delivers it at time 15.

2) Before checking G on Bp, we need to make sure whether the process may take place successfully, i.e. the preconditions of each action are satisfied when it is evoked and the constraints are satisfied too. Especially, when the process is proposed as future activity. Formally, it is confirmed if at least one interpretation $\mathcal{I}$ exists where i) the only actions that may occur are occurs(A,T1,T2) in Bp (to exclude intervention of other actions); ii) $\mathcal{I}(Bp) = \text{true}$. We simply establish here it is so with Bp. Continue to 3). If such interpretation doesn't exist, this means the process can not be performed in any way. Go to 1).

3) Check whether the formula

$$\text{impl}(G) = \text{stop\_supply\_after}(30)$$

implementing the goal is true in the interpretation $\mathcal{I}$ as well. A yes means, the activities in Bp achieve the goal and may be performed as they are. This is a desirable situation. However, if the formula is not true in $\mathcal{I}$, which is the case with stop_supply_after(30), this means Bp violates G.

In this way we can formally inspect whether goals are/will be achieved in current/future business processes, and give an answer to "How can we check whether goals are achieved or not in real business processes?". At the same time goals are used as *guiding principles for business activities*. That is, when a process turns out to be violating some goal, it can be re-examined concerning the goal. For example, we may find out why the goal G above is not satisfied by the process Bp by tracing the reasons why $\mathcal{I}(\text{impl}(G)) = \text{false}$. Based on the reasons we may even find out how to improve the business process so that it is still performable but satisfies the goal. For example, remove the action occurs(supply(ag3, ag1, 1002, p2, i1), 53, 59) from Bp, meaning that ag3 does not supply. Therefore the question "If the goals are not satisfied, why and what to do?" can be answered. On the other hand, it can be shown that the goal prompt_delivery is achieved in Bp, which means that the process is acceptable w.r.t. the goal.

Yu & Mylopoulos [8] have also suggested to reason with goals. However, their reasoning is done in an informal way and the main purpose with the reasoning is to examine alternative ways to achieve (functional or non-functional) goals in the context of business process reengineering, rather than to check whether goals are attained in business processes.

4 Final Remarks

In this paper we have presented a formal approach to representing and reasoning with goals. Our message was that it is not enough with representing the goals as an essential part of a business enterprise: There should be some systematic way to investigate whether the goals are attained in reality. We have demonstrated how goal satisfaction can be studied by formal reasoning with business processes.

However, much work needs to be done in the future. Goal structure needs to be analysed deeper to consider, e.g. deontic aspects [4], and conflicts between goals. The language should support methods for structuring complex concepts as pointed out by, e.g. [5]. Especially, in order for our approach to be applied in practice, for example, supporting automatic reasoning with business processes, the semantics of the logic should be implemented in some effective algorithms.

References

[1] P. Assenova and P. Johannesson. First Order Action Logic- an Approach for Modelling the Communication Process between Agents. 1st Intl. Workshop on Communication Modeling - The Language/Action Perspective, Eds. F. Dignum, J. Dietz, E. Verharen and H. Weigand, electronic Workshops in Computing, Springer-Verlag, 1996.

[2] A. Dardenne, Van Lamsweerde and S. Fickas. Goal-directed requirements acquisition. *Science of Computer Programming* 20, pp. 3-50, 1993.

[3] E. Dubois. A Logic of Action for Supporting Goal-Oriented Elaborations of Requirements. *ACM SIGSOFT Software Engineering Notes*, Vol. 14, No. 3, pp. 160-168, 1989.

[4] J. Krogstie and G. Sindre. Utilizing deontic Operators in Information Systems Specification. *Requirements Engineering* 1:210-237, 1996.

[5] J. Mylopoulos, A. Borgida, M. Jarke and M. Koubarakis. Telos: Representing Knowledge About Information Systems. *ACM Transactions on Information Systems*, Vol 8, No. 4, pp. 325-362, 1990.

[6] Erik Sandewall. Features and Fluents, A Systematic Approach to the Representation of Knowledge about Dynamical Systems, Oxford University Press, 1994

[7] Choong-ho Yi. PhD thesis manuscript in preparation.

[8] E. Yu and J. Mylopoulos. Using Goals, Rules and Methods to Support Reasoning in Business Process Reengineering. Proc. of the 27th Annual Hawaii Intl. Conf. on Systems Sciences, Vol. 4, pp. 234-243, 1994.