Goals, which can be described as states that an agent would like to realize, is an important concept for intelligent agent systems. The representation of goals and the ability to reason about them are the major problems in goal-oriented analysis and modeling techniques, especially in intelligent agent systems, as goals are more stable than other abstractions (e.g., user stories). Description Logics (DLs) is a formal tool of knowledge representation and reasoning. In this paper, we construct a framework with explicit representation and formal semantics of goals—Goal Description Logics (GDLs), which integrates two aspects of goals: declarative (a description of the state of sought), and procedural (a set of plans for achieving the goal), into one concept based on Description Logics (DLs). In addition, goals reasoning, especially goal matchmaking in GDLs, is studied using its effective judgment to concept subsumption. We propose a conceptual model of goal-based migration workflow system (GMWfS) based on GDLs, and illustrate an application. We also present preliminary experimental results on an implementation of these ideas. Compared to traditional workflow methods, GMWfS is more flexible and intelligent.
Description Logics (DLs) are a family of knowledge representation formalisms. They are based on the notion of concepts and roles, and are mainly characterized by constructors that allow complex concepts and roles to be built from atomic ones (Horrocks & Sattler, 1999; Baader, Calvanese, Mc Guinness, Nardi, & Patel-Schneider, 2002). The main benefit from these knowledge languages is that sound and complete algorithms for the subsumption and satisfiability problems can be defined. A DLs reasoner solves the problems of equivalence, satisfiability and subsumption.

A key difference between DLs and the standard representation formalisms based on First-Order Logic, e.g. relational and deductive databases, is that DLs provide an arena for exploring new sets of “logical connectives” – the constructors used to form composite descriptions – that are different from the standard connectives such as conjunction, universal quantifiers, etc. Therefore, DLs provide a new space in which to search for expressive and effectively computable representation languages. Moreover, although it is possible to translate many aspects of expressive and effectively computable representation languages is that sound and complete algorithms for the subsumption and especially its effective judgment to concept subsumption. The main contributions of this paper include:

- A logic framework to express goals description in intelligent agents, including syntax and semantics;
- Reasoning, especially goal hierarchy construction algorithms in Goal Description Logic (GDLs);
- A conceptual model of goal-oriented migration workflow system (GMWS);
- Evaluation of goal-based workflow system and process-based workflow system.

The remainder of the paper is structured as follows: in the next section (Section 2), we define a framework of goals and plans-Goal Description Logics (GDLs). In Section 3, we present goal hierarchy algorithms, which is an important reasoning problem in GDLs. Then, in Section 4 we propose a goal-based migrating workflow model, and report experiments to show the costs and benefits of goal-based workflow system compared to traditional workflow system. Section 5 reports on background work and the related work, to finally draw some conclusions and propose some further research topics in Section 6.

2. Goal description logics (GDLs)

In order to implement goal representation and reasoning, we propose a formal framework-Goal Description Logics (GDLs) based on Description Logics (DLs). In this section, we will present the formal syntax and semantics of GDLs, and then introduce the main means of goals achievement-goals plan.

Let’s suppose Cons to be a constant set: Cons = \{a_1, a_2, \ldots\}; Var is a variable set: Var = \{x_1, x_2, \ldots\} and Fun is a function set: Fun = \{f_1, f_2, \ldots\}.

2.1. Syntax

**Definition 2.1** (Term). A term in GDLs can be defined as follows:

1. If \( x \in \text{Var} \), then \( x \in \text{Term} \);
2. If \( x \in \text{Term} \) and \( x_1, x_2, \ldots x_n \in \text{Term} \), then \( f(x_1, x_2, \ldots x_n) \in \text{Term} \).

**Definition 2.2** (Concept).

1. Primitive concept \( P \), top concept \( \top \) and bottom concept \( \bot \) are concepts;
2. If \( C \) and \( D \) are concepts, then \( \neg C \), \( C \cap D \) and \( C \cup D \) are concepts;
3. If \( R \) is a role name, and \( C \) is a concept name, then \( \exists R.C \) and \( \forall R.C \) are concepts.

**Definition 2.3** (Formula).

1. Assertion formula like \( C(a) \), \( R(a, b) \), where there is no variables;
2. Common formula like \( C(x) \), \( R(x, y) \), where there is at least one variable;
3. Assertion formula and common formula are formulae;
4. If \( \varphi \) and \( \psi \) are formulae, then \( \neg \varphi, \varphi \land \psi, \varphi \rightarrow \psi \), are all formulae.

**Definition 2.4** (Substitution). A substitution \( \sigma \) is a finite set of pairs as the form \( \{a_1/x_1, a_2/x_2, \ldots a_n/x_n\} \), where \( a_i \in \text{Var} \) and \( x_i \in \text{Term} \), and \( \forall i \neq j: a_i \neq a_j, i, j \in [1 \ldots n] \). A ground substitution \( \sigma \) is a substitution such that there is no variable in \( x_i \).

**Definition 2.5** (Instance Formula). Let \( \phi \) be a GDLs formula, \( \theta = \{a_1/x_1, a_2/x_2, \ldots a_n/x_n\} \) be a substitution. Then \( \phi^\theta \) is called an instance formula where all variables in \( \phi \) are simultaneously replaced with \( \theta \).

**Definition 2.6** (Atomic Goal). An atomic goal in GDLs can be described as follows: \( g(x_1, x_2, \ldots x_n) = (P, S, F) \), where:

1. \( g \) is the atomic goal name. \( x_1, x_2, \ldots x_n \) are constants or variables;
2. \( P \) is the pre-condition set which is made up from \( C(x_1) \) or \( R(x_i, x_j) \) (\( 1 \leq i, j \leq n \)). \( P \) guarantees that the goal \( g \) can be satisfiable. In a general way, we call \( P \) satisfaction conditions;
3. \( S \) is the state of the world once a goal succeeds. \( s \in S \) has the form as \( B/H_i \), where \( B \) is a set of GDLs formulae; \( H_i \) is a common literal, having the form as \( \neg C(x) \), \( \neg R(x_i, x_j) \), where \( C \) is a concept and \( R \) is a role. If \( H_i \) is positive, then \( H_i \) is added to \( Abox.A \); else \( H_i \) is deleted from \( Abox.A \);
4. \( F \) is a set of failure conditions which indicates a goal must be dropped with failure. One advantage of allowing \( F \) to be specified is that it allows for a range of commitment strategies to be realized by specifying appropriate conditions for a goal to be dropped.

Given an atomic goal, we can get a goal instance by a substitution \( \theta \). A ground goal is a goal without any variable. An atomic goal can be regarded as a set of ground goals.

**Definition 2.7** (Goal Formula). A goal formula (goal) can be constituted by:

1. If \( g \) is a n-dimension atomic goal name, \( t_1, t_2, \ldots t_n \) are terms, then \( g(t_1, t_2, \ldots t_n) \) is a goal formula, called atomic goal formula;
دریافت فوری
متن کامل مقاله

امکان دانلود نسخه تمام متن مقالات انگلیسی
امکان دانلود نسخه ترجمه شده مقالات
پذیرش سفارش ترجمه تخصصی
امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
امکان دانلود رایگان ۲ صفحه اول هر مقاله
امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
دانلود فوری مقاله پس از پرداخت آنلاین
پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات