Human-Machine Collaboration over Linked Data

Paolo Pareti *

University of Edinburgh, Edinburgh, United Kingdom
Taiger, Madrid, Spain
paolo.pareti@taiger.com

Abstract. This study presents a framework to allow human and machine agents to reason and coordinate actions without direct communication mechanisms by sharing distributed Linked Data resources. This framework addresses the problems of querying frequently-updating distributed datasets and guaranteeing consistency in case of concurrent updates. The motivation for this framework comes from the use-case of opportunistic automation of humans-generated procedures. This use-case is based on existing real-world Linked Data representations of human instructions and their integration with machine functionalities.

1 Introduction

On the web, the amount of structured information available to machines is steeply increasing. This information can be used by computer systems to answer complex queries about factual knowledge, for example about the population of cities and the date of birth of notable people. However, machines still have little or no understanding of human activities. Achieving a complex task often requires using different software tools. However, these systems might operate in isolation, not knowing what the user is trying to achieve and how. This lack of understanding is a limitation to human-machine collaboration as machines cannot predict when and how their functionalities might be needed.

A typical approach to describe activities is by writing instructions. Previous research has demonstrated how human instructions can be converted into Linked Data and how related tasks and entities can be automatically interlinked [2]. The feasibility of this process was demonstrated by the creation of an RDF dataset[1] of over 200.000 procedures extracted from the wikiHow[2] and Snapguide[3] websites. While certain steps of the instructions can only be executed by humans, others, such as sending emails or modifying files, can be automated. Such steps can be linked to machine functionalities and executed at the right time when a user is performing a related activity [1]. This framework generalises this approach to

* This research has been funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 607062 ESSENCE: Evolution of Shared Semantics in Computational Environments.

[1] http://w3id.org/knowhow/dataset

[2] http://www.wikihow.com/

[3] http://snapguide.com/

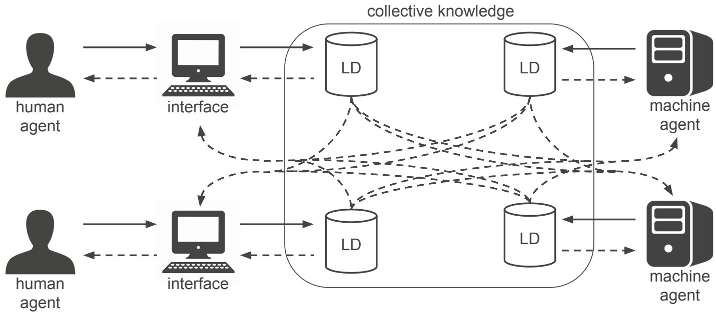

Fig. 1. A multi-agent environment with no shared resources. Dashed and solid lines represent, respectively, the ability to read and write Linked Data resources.

automation to the case of multiple agents and decentralised resources. Given the high frequency of updates of the Linked Data resources used for communication, this scenario presents new challenges to traditional distributed query approaches. Also, the potentially small size and decentralisation of these resources make it an ideal application scenario for dynamic Linked Data discovery at runtime.

2 Problem Description

The proposed framework addresses the problem of allowing a collaborative set of human and machine agents, who can publish and access web resources but that cannot directly interact with each other, to *communicate* and *coordinate* their actions to collaboratively achieve tasks. As depicted in Figure 1, agents can access the resources of the other agents, but can only modify their own. No centralised resource which multiple agents can modify is available. In this context, *communication* refers to the process by which agents can propagate information (i.e. triples) to the other agents by modifying the *collective knowledge*, namely the resources that all agents can access. *Coordination* instead refers to the ability to guarantee certain conditions across all datasets. For example, coordination might be required to ensure that no agent starts executing a task which is already being executed by another agent. To simplify and isolate the problem at hand it is assumed that the agents involved already know and trust each other. Issues such as agent discovery, coalition formation and trust are considered as outside the scope of this project.

3 Framework Description

3.1 Knowledge Representation

In order communicate meaningfully, a shared knowledge representation format needs to be established. Following the Linked Data principles, agents represent their knowledge as RDF statements and agree on a shared vocabulary. If this is not the case, additional techniques could be used. For example, knowledge extraction tools could generate an RDF representation of unstructured data and ontology alignment tools could be used to make different vocabularies interoperable.

One of the main challenges to enable communication between humans and machine agents is the representation of knowledge in a format which is both human and machine understandable. It is therefore important to map such representation on the one hand to a logical formalism, so as to allow machine reasoning, while on the other hand to an intuitive representation, such as in natural language, which can be understood by humans. In the human-machine collaboration scenario the PROHOW[4] vocabulary is adopted. This vocabulary represents tasks in terms of instructions and of their execution and it can be translated both into a natural language representation and into logical statements [1].

3.2 Communication

Having decided on a shared knowledge representation format, agents communicate by storing data in their own repositories. The preferred method of doing so is by using dereferencable URIs. For example, the URI for a certain task could resolve to an HTML document containing human understandable instructions on how to achieve the task, along with embedded machine understandable RDFa data. Content negotiation can also be used to serve human understandable documents to human users, and data files, such as RDF/XML data, to machine agents.

Dereferencable URIs allow users to retrieve human understandable representations of entities using web browsers. More complex human interactions, however, require dedicated applications, here called *interfaces*. Human agents interact with the collective knowledge through interfaces which allow them both to consume and generate Linked Data. Both interfaces and machine agents can retrieve the collective knowledge of all the other agents locally to reason over it. In order for this to happen, agents need to know the URIs of the datasets of the other agents. Knowledge of whether the others agents are humans or machines, or how to interact with them directly, is not required. In order to keep the collective knowledge up to date, agents need to locally retrieve the datasets of the other agents on a frequent basis, since datasets are constantly updated as the state of the collaboration evolves. This fact imposes practical constraints on the amount of data that agents can share during their collaboration.

Interfaces also require server-side functionalities in order to publish and update Linked Data. For example, a human agent might use an interface to describe how a certain task can be completed. An interface might then translate the user input (e.g. natural language instructions) into RDF and then publish it online. Interfaces should also translate user actions, such as a click on a check-box to define a step of a procedure as "complete", into corresponding RDF statements.

3.3 Coordination

Coordination between agents might require them to verify certain conditions before a statement is communicated. For example, an agent a collaborating with

[4] http://w3id.org/prohow

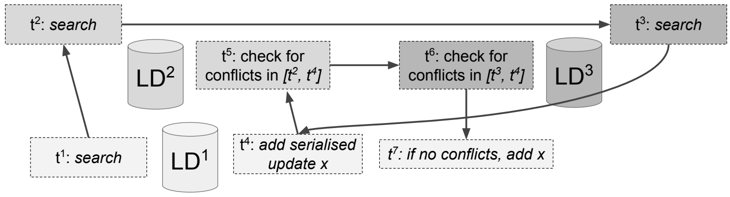

Fig. 2. A simple coordination mechanism for distributed datasets.

two other agents might be allowed to declare its intention s to accomplish a certain task only under a condition c that no other agent is already planning to do so. A possible approach to ensure such conditions is illustrated in Figure 2. Agent a might start by retrieving Linked Data from all the knowledge bases to determine if condition c is met (timepoints t^1 to t^3). Even if the condition appears to be satisfied, agent a cannot declare statement s yet as the other agents, in the meanwhile, could have updated their own datasets and invalidated c. The proposed approach to solve this problem is for agent a to write statement s in a "provisional" format (timepoint t^4) using RDF reification, exploiting the fact that reified triples do not entail the triples. Reification provides a unique identifier for each triple, which can be annotated with the timestamp of its creation. Agent a would then access the other agent's knowledge bases a second time to verify if condition c still holds (timepoints t^5 and t^6). The agent will also consider the provisional statements written by the other agents before s was added in provisional form (before t^4) but ignore the ones written after. After accessing the collective knowledge base a second time, agent a will add statement s in its repository only if condition c still holds (timepoint t^7). This approach ensures that conflicting statements are written on a first-come first-served basis.

4 Conclusion

This work presents a framework to allow human and machine agents to collaborate without directly communicating with each other. Linked Data is used as an indirect communication mechanism where agents communicate by publishing and accessing distributed resources. Issues arise as resources used to collaborate need to be frequently updated. Approaches are discussed to ensure that updates are quickly propagated to all the agents and that concurrent ones do not violate global constraints. Knowledge about processes and their execution is represented in an existing format which is both human and machine understandable.

References

1. P. Pareti, E. Klein, and A. Barker. Linking Data, Services and Human Know-How. In *The Semantic Web. Latest Advances and New Domains*, volume 9678 of *LNCS*, pages 505–520. 2016.
2. P. Pareti, B. Testu, R. Ichise, E. Klein, and A. Barker. Integrating Know-How into the Linked Data Cloud. In *Knowledge Engineering and Knowledge Management*, volume 8876 of *LNCS*, pages 385–396. 2014.