Engineering Applications of Artificial Intelligence 35 (2014) 215–236

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Situation-aware composition and execution in dynamic environments by automated planning

Qiang Lu [a,b,*], Justin Wilson [c], Yixin Chen [c], Christopher Gill [c], Louis Thomas [c], Gruia-Catalin Roman [d], Guoliang Chen [b]

[a] College of Information Engineering, Yangzhou University, 196 Huayang Road, Yangzhou, Jiangsu, 225127, China
[b] School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230026, China
[c] Department of Computer Science & Engineering, Washington University in Saint Louis, St. Louis, MO 63130, USA
[d] School of Engineering, The University of New Mexico, Albuquerque, NM 87131, USA

ARTICLE INFO

Article history:
Received 21 August 2013
Received in revised form
25 June 2014
Accepted 29 June 2014
Available online 24 July 2014

Keywords:
Workflow
Mobile environments
Coordination
Automated planning

ABSTRACT

Existing workflow management systems execute complex sequences of tasks by invoking corresponding services identified at execution time and ordered according to a graph that captures the workflow execution rules and data dependencies. The technology is mature and is used across the Internet to support a wide range of business transactions by taking advantage of a relatively stable communication and service infrastructure. This paper extends the applicability of workflow technology by considering settings in which service availability is highly variable due to the mobility of people and devices and due to frequent and extended disconnections from the wired infrastructure. This paper extends workflow technology to support goal-directed coordination among physically mobile people and devices that form a transient community over a mobile ad hoc network (MANET). Our approach is based on the notion of open workflows, i.e., workflows that are constructed dynamically using available knowledge and services. Because the workflow construction and execution are situation aware, different workflows are built under different circumstances even when the need being addressed is unchanged. We also introduce a framework that co-designs workflow allocation and host scheduling in a unified way, which can handle communication range limitations in a mobile environment while previous methods cannot. Specifically, we transform the collaboration problem into a temporal planning model and then solve it using automated planners. We integrate this framework into the CiAN workflow management system, which was developed in our previous work. Results of experiments we have conducted show that our approach significantly broadens the scope of previous research by solving the challenges of *availability*, *mobility*, and *communication constraints* in MANET domains.

1. Introduction

As the increasing usage of small, powerful wireless devices, computing will embrace the frequent, transient, ad hoc interactions of mobile environments. Since computing and communication become more and more integrated into the fabric of our society, there will be large demands for new kinds of enterprises and new forms of social interactions. In this paper, we propose a new framework to support arbitrary collaborations among groups of people (and their personal devices) in a mobile ad hoc network (MANET). Application domains that motivate or even require this form of interaction include low profile military operations, emergency responses to major natural disasters, scientific expeditions in remote parts of the globe, field hospitals, and large construction sites. These application domains exhibit several key features: ad hoc interactions among people, high levels of mobility, the need to respond to unexpected developments, the use of locally available resources, prescribed rules of operation, and specialized knowhow.

The coordination problem in this MANET environments usually results in how to arrange a series of activities or tasks according to a specific order to achieve a high level goal, which can be seen as a workflow that is reactive, opportunistic, composite, and constrained by the set of participants present on the site along with their knowledge and resources. Current workflow middleware allows people to initiate complex goal-oriented activities that leverage services made available by a wide range of service-oriented portals. In a typical scenario, a user employs a web browser

* Corresponding author at: College of Information Engineering, Yangzhou University, 196 Huayang Road, Yangzhou, Jiangsu, 225127, China.
E-mail address: lvqiang@yzu.edu.cn (Q. Lu).

http://dx.doi.org/10.1016/j.engappai.2014.06.024
0952-1976/

to make a request to a workflow engine responsible for executing a predefined workflow that can satisfy the specific user need. While the workflow itself is fixed and defined ahead of time, its execution is adaptive as services are selected at runtime through a discovery and matching process facilitated by the service specifications that are part of the workflow and by the service directories and service providers.

However, consider the new features in MANET domains, the stability assumptions under which such workflow technology operates need to be reexamined. In many situations, people and devices are agents that deliver services interacting with the physical world. This leads to an extended view of what a service is and induces a high level of variability into the system. People are not always ready to help and, similarly, devices can be busy, exhibit independent behavior, or be subject to environmental controls. Generally speaking, different hosts have different capabilities, availability, and other resources. In a dynamic community which members can join and leave at will in a MANET environment, there is no way to know which resources can be used before the participants have been fixed. Thus, there will be no fixed server that can store all resources. In such an environment, the availability of resources is distributed across all participants.

Designing a workflow management system (WfMS) for these domains faces many challenges, as hosts may move and service availability may depend upon which hosts are within communication range. To extend the use of workflow technology to these highly dynamic environments, it is also important to consider the idea of building workflows on the fly in response to a specific need being expressed in a particular setting. At one extreme one might consider for each user the availability of a fixed set of workflows among which one is selected automatically based on existing resources and other considerations. At the other extreme one could employ sophisticated automatic reasoning systems to construct a workflow totally from scratch. Our approach assumes that knowledge about how to respond to certain basic needs is available in the form of small *workflow fragments* distributed across a set of hosts or accessible from servers. Workflow fragments are merely small workflows (possibly even a single task) that are intended to be composed into larger workflows at a later time. As new and unanticipated needs are identified complex workflows are constructed by composing existing workflow fragments in a way that pays attention to the availability of resources across the relevant spatiotemporal domain.

These challenges make most existing workflow management algorithms inadequate since they do not consider the mobility and communication constraints in the ad hoc settings. Standard workflow management systems, such as ActiveBPEL (Active Endpoints, 2008), Oracle Workflow Engine (Oracle Inc., 2007), JBoss (JBoss Labs, 2007), and BizTalk (Microsoft Corp, 2007), are designed to work in fully wired environments, such as corporate LANs or across the Internet. Reliance on centralized control and reliable communication means that such solutions cannot robustly operate under the constraints of dynamic mobile environments. The work on federating separate execution engines running independent workflows (Omicini et al., 2006) removes the requirement of centralized control. Kumar et al. (2004) investigate decentralized orchestration of a single workflow by partitioning the workflow at build time and using message passing at run time. However, both approaches still assume reliable communication and a fixed group of hosts. A closely related WfMS is WORKPAD (Mecella et al., 2006), which was designed to support collaborative work of human operators in emergency/disaster scenarios. WORKPAD requires a stable wireless connected community where all devices would not leave or join in and at least one member of the MANET to be connected with a central coordinating entity that orchestrates the workflow and shoulders any heavy computational loads. While in MANET domains the workflow is generated on the fly to match the present situation, all above related systems assume that a fully specified workflow already exists. A related work which supports dynamic workflows is Worklets (Adams et al., 2006). It adopts a service oriented architecture which supports flexible work practices based on accepted ideas of *Activity Theory* which focuses on understanding human activities and work practices (Nardi, 1996).

Some recent research efforts targeted toward the MANET domains (Sen et al., 2007, 2008; Thomas et al., 2009; Haitjema et al., 2010) still confront some limitations. For example, CiAN adopts a simple heuristic allocation algorithm which divides the workflow allocation into sub-problems recursively (Sen et al., 2007, 2008). Instead of using a recursive allocation algorithm, Haitjema solves this problem by transforming it to a numeric temporal planning problem and calling a temporal planner to find a feasible allocation (Haitjema et al., 2010). These works de-couple workflow allocation from host scheduling which largely simplifies the workflow problems. However, they may fail to find a feasible solution that supports a given workflow allocation due to the simplification, even if there exists a feasible solution under another workflow allocation.

In our previous work (Thomas et al., 2009), we addressed this problem by dividing it into two independent phases: constructing a feasible workflow plan and allocating all tasks in the workflow plan to hosts. The workflow construction phase first builds a workflow graph which consists of all feasible workflows and then selects one feasible workflow by using an iterative pruning algorithm. Then in the allocation phase, it simply determines how and to whom to distribute each task in the constructed workflow according to each host's task availability information. When it assigns a task to a host, it allows the host to decide independently how and when to move to the task specific location, when to execute the task, and how to exchange data and coordinate with other hosts. Splitting the original problem into these two phases largely simplifies the planning problem. However, it makes two assumptions: (1) the constructed workflow plan can be successfully allocated and (2) all hosts will find feasible movement schedules to satisfy the host communication constraints. In real-world applications, especially in dynamic MANET environments, such assumptions may be violated because they do not consider the host movement and communication constraints when allocating the workflow. In other words, by de-coupling workflow allocation from host scheduling, the approach may fail to find a feasible solution that supports a selected workflow allocation, even if there exists a feasible solution under another workflow allocation solution. Even if all tasks can be allocated in the first phase, it still cannot guarantee that two hosts are in communication range at a future time when they need to communicate with each other. Moreover, considering the feasible workflow plan, workflow plan allocation, and host schedule together gives a larger decision space which may lead to more preferable (e.g. shorter) plans. The *co-design* of workflow plan, workflow allocation, and host scheduling, which considering the dependencies, communication, and temporal constraints, is beyond the capability of the allocation algorithm in CiAN, or any other existing workflow algorithms we know of.

In order to address these challenges for collaboration in MANETs, in this paper we extend the open workflow scheme introduced in our previous work (Thomas et al., 2009). Fig. 1 gives an overview of our co-design open workflow (COW) scheme and our previous open workflow scheme. In these schemes, we both first use the open workflow scheme to construct workflow problems dynamically. Specifically, once a new task requirement is given, we first collect relevant knowledge from participants and construct a workflow problem represented by a graph. After that,