EFFICIENT COLLABORATIVE SCIENTIFIC WORKFLOW COMPOSITION USING SERVICE ORIENTED COLLABORATION MODEL

SUSHMA. K¹, MANJULA.T², LAVANYA.K³

1, M.Tech (CSE), PBRVITS, KAVALI.
2, Associate Professor, CSE,PBRVITS, KAVALI.
3, M.Tech (CS), PBRVITS, KAVALI.

Abstract: Collaboration has become a dominant feature of modern science. Many scientific problems are beyond the realm of individual discipline or scientist to solve and hence require collaborative efforts. Modern scientific data management and analysis usually rely on multiple scientists with diverse expertise. In recent years, such a collaborative effort is often structured and automated by a dataflow-oriented process called scientific workflow. Such workflows may have to be designed and revised among multiple scientists over a long time period. Existing tools are single user-oriented and do not support workflow development in a collaborative fashion. Based on scientific collaboration ontology, we propose a service-oriented collaboration model supported by a set of composable collaboration primitives and patterns. The collaboration protocols are then applied to support effective concurrency control in the process of collaborative workflow composition. The Design and development of Confucius a service-oriented collaborative scientific workflow composition tool that extends an open-source, single-user environment

Index Terms— H.4.1.g Workflow management, M.4 Service-Oriented Architecture, H.5.3.c Computer-supported cooperative work, Confucius tool

I. INTRODUCTION

The advancement of modern science has created sheer volume of data with increasing complexity. Processing and managing such large-scale scientific data sets is usually beyond the realms of individual scientists to solve; instead, it has to rely on multiple domain scientists with diverse expertise. For example, the Large Synoptic Survey Telescope (LSST) experiment, which aims to repeatedly image half of the sky over a planned 10-year survey, produces data at a rate of 300 MB/s and will result in catalogs of about 130 TB of roughly $3 \times 10^9$ sources times 10 years worth of data. Analyzing such data sets demands a collaboration of a number of organizations with over 1,800 scientists and engineers engaged. Such scientific data analysis and processing is usually structured and automated by a dataflow-oriented process called scientific workflow. In contrast to business processes that are control-flow oriented and orchestrate a collection of well-defined business tasks to achieve a business goal, scientific workflows are often dataflow-oriented and streamline a collection of scientific tasks to enable and accelerate scientific discovery. Researchers use scientific workflows to integrate and structure local and remote heterogeneous computational and data resources to perform in silico experiments.
implies that the accuracy of the entire process is determined by the design of each step and the error propagation between them. While involving scientists with different capabilities focus on finding a local optimal design of a particular step, collaboration between them will find a global optimal design for the entire workflow. We focus on reporting our design and development of one key enabling technique, collaboration protocol. We propose scientific collaboration provenance ontology, and based on it, we have developed a service-oriented collaboration model that is supported by a set of composable collaboration primitives and patterns. The collaboration protocols are then applied to support effective concurrency control in the process of workflow co-design. The core idea of Service Oriented Architecture (SOA) is to position services as the primary means (components) that encapsulate solution logic as reusable assets. In this project, we have leveraged the concept of SOA to build our system for higher interoperability, reusability, and productivity. Since we focus on scientific workflows, throughout this paper, we use the terms scientific workflows and workflows interchangeably.

2. RELATED WORK

2.1 Scientific Workflow Management Systems

To date, several scientific workflow management systems (SWFMSs) have been developed as single-user environments that help individual scientists construct workflows from available scientific resources. Representative SWFMSs include Kepler, Taverna, Triana, VisTrails, Pegasus, Swift, Trident, and VIEW. Each system shows unique features. Each of the SWFMSs provides a platform to support individual scientists in composing. Some systems show some collaboration features, in the sense that they allow a scientist to compose a workflow from shared resources and services. However, they provide limited support for multiple scientists to collaboratively compose a shared workflow.

2.2 Business Process Coordination

For business workflows, the term "collaborative workflows" is interchangeable with the term "coordinated workflows". They emphasize coordination between workflows. The Computer Supported Cooperative Work (CSCW) community has studied the general-purpose concurrent design problem, where collaborators with different ownerships possess different controls over the shared work product. To name a few, OntoEdit supports a collaborative software engineering process; Yen et al. present a collaborative design tool that allows privileged collaborators to change the process; OPCA Team integrates the object-oriented and process-oriented paradigms into one single framework to enable the coexistence of structured processes and human interaction behaviors in one business process modeling system. In contrast, our work focuses on dataflow-oriented collaboration, where semantic relationships and constraints between different comprising components (e.g., tasks and data links) need to be carefully considered during concurrent composition.

2.3 Collaborative Workflow Composition

We surveyed the state of the art of the field of scientific workflows toward the support of collaborative scientific workflows. Our observations directly motivated the research work reported in this paper. We also have surveyed the literature of workflow control mechanisms in a collaborative environment in. Our study helped us build a linkage between scientific workflow and collaborative work. Our “collaboration provenance” is different from the term “collaborative provenance” used by Dr. Altintas etal. Their “collaborative provenance” means, “inferring dependencies across multiple workflow runs and understanding user collaborations based on scientific workflow runs”. In contrast, our term “collaboration provenance” aims to capture how scientists collaboratively design a common scientific workflow, e.g., who has designed which part.

3. COLLABORATIVE SCIENTIFIC WORKFLOW COMPOSITION MODEL

Provenance has been widely considered critical to the reproducibility of scientific workflows. Compared to existing significant amount of work focusing on provenance for runtime workflow execution, our work focuses on collaboration provenance that tracks human interactions and efforts in the process of scientific workflow composition. Our method is to record all collaborative activities that contribute to a composed workflow.

3.1 Provenance Ontology

We have developed provenance ontology to support the modeling of various provenance data
recording scientific workflow design and user interactions during the process of a collaborative workflow design. As shown in Fig. 1, our ontology is centered upon the concept of “workflow.” Each scientific workflow comprises organized processors (tasks) and data links (aka. data channels), sub-workflows, as well as predefined requirements and annotations (comments). Each workflow maintains one or more floors that are tokens to ensure concurrency control. Long-term collaboration on a scientific workflow forms a meeting. A short-term synchronous collaboration is called a session. Each scientific workflow belongs to a project. Each project belongs to a scientific group (could be a virtual group). Each group may comprise multiple focuses, each involving multiple projects. A group contains a set of members, each may belong to different organizations.

Fig.2: Collaborative workflow composition
Provenance ontology

3.2 Collaborative Composition Model

We designed models to regulate how collaborators can collaboratively design and update mutual workflows. Instead of reinventing the wheel, we chose to explore how to extend the single user-oriented Taverna tool.

3.3 Advanced collaboration model

Scientific collaborations usually last for a long period of time, e.g., months and years. In addition, temporary discussion groups and sessions may be formed in the lifecycle of a long-term scientific collaboration process. Therefore, we constructed a hierarchical structure for the central server. It may host multiple collaboration groups, which may or may not have nesting relationships between them. The central server maintains all collaboration group information and acts as the subject for all registered groups. All observers (collaborators) are organized into corresponding collaboration groups. The central server also stores and manages all provenance data, so that it becomes a repository of workflow products and enables scalability. In other words, we realize a multi-tenancy infrastructure.

Fig.3: Collaborative composition model.

4. WORKFLOW COMPOSITION CONCURRENCY CONTROL

At composition time, multiple scientists collaborate to develop a scientific workflow. Thus, concurrency control deserves consideration to ensure design productivity.

4.1 Locking Granularity

Adopting the instrument from an extensively tested and well proved human communication protocol, RRO, we originally established a workflow-level floor control. Each collaborator competes for the shared floor before making any changes. Such a workflow-level floor control may not be efficient to support large-scale scientific workflow composition, though. Since scientific research is an exploratory process, the development of a workflow may undergo many rounds of discussions and changes and may last for a long period of time. Meanwhile, a collaboration group nowadays usually comprises scientists from different organizations at distributed locations. They may possess different schedules and may even reside in different time zones; thus, their collaboration may adopt both synchronous and asynchronous modes. Furthermore, a large scale scientific workflow may involve many comprising components. It is neither efficient nor practical, if one scientist working on one component locks the
entire workflow and prevents other scientists from working on unrelated components.

![Word count workflow diagram]

Fig. 4: Word count workflow.

If we set up the locks on individual processors and data links only, two collaborators may concurrently update one processor Mapper1 and its output data links, respectively. This situation may not be desirable, because the data link directly depends on the processor. In other words, connected processors and data links may have close semantic relationships, which need to be preserved by requiring that adjacent entities cannot be updated by different collaborators at the same time.

4.2 Locking Algorithms

Based on the concept of synchronization area, we developed four algorithms, on locking/releasing a processor and locking/releasing a data link. If a user selects a processor to lock it, we first check whether it has been locked by another collaborator. If nobody locks it, then an uninterruptable transaction starts. First, we set the lock flag of the processor, and fill the name of the owner of the processor. For each outgoing data link of the processor, we check whether there is an active lock on it. If any outgoing data link is currently locked by other collaborators, the entire locking attempt will be aborted.

Algorithm 1: Floor Granting Algorithm

**Input:** A collaborator requests a floor

**Requirements:** Release a floor.

1. check(waiting_list)
2. if (waiting_list ≠ ∅) then
3.  requestor = get_top_requestor(waiting_list)
4.  floor_owner = requestor
5.  notify(requestor)
6.  remove(requestor, waiting_list)
7.  else if (waiting_list = ∅) then
8.  floor_flag = unoccupied
9.  notify(requestor)
10: endif

Algorithm 2: Floor Releasing Algorithm

**Input:** A requestor requests a floor

**Requirements:** Decide whether a floor should be granted.

1. check(floor)
2. if (floor ≠ taken), then
3.  floor_flag ← occupied
4.  floor_owner ← requestor
5.  notify(requestor)
6.  return true
7.  else if (floor = taken) then
8.  insert(requestor, waiting_list)
9.  return false
10: endif

Algorithm 3: Locking algorithms

6. SYSTEM IMPLEMENTATION

We have constructed a collaboration pattern template library. The basic building blocks are collaboration primitives. Users can build new collaboration patterns using existing collaboration primitives. Identified collaboration patterns are stored as provenance data to support the tracking, storing, and querying of interactions and coordination among scientists. Without reinventing the wheel, we extended the single user-version Taverna into a collaborative version. The reason why we chose Taverna is mainly based on its popularity and large user base. Another reason is that Taverna is an open-source tool developed in Java. Thus we can explore its code base.

We built a central server supporting all workflow collaborations. Workflow evolution provenance and collaboration provenance are stored in a shared database on the server. Each collaborator may store an intermediate version of the workflow at a local machine, but all committed activities are stored at the server, to support asynchronous collaboration where collaborators may work on the shared workflow at preferable
We consider four options for selecting database systems: native XML, relational, XMLrelational, and RDF. Currently we use a relational database because it is a preferable choice of Taverna, upon which our Confucius is built.

Fig. 5: Screen shots of concurrent workflow updates

7 CONCLUSIONS

With the advent of ongoing work on establishing collaboration protocols to support collaborative scientific workflow composition, Our service-oriented infrastructure includes a collaboration ontology associated with a set of collaboration patterns, primitives, and constructs, as well as concurrent control mechanisms to support concurrent collaborative workflow composition. We plan to continue our research in the following directions. First, we will design and conduct an evaluation study and use the feedback to enhance the system. Second, based on the collaboration ontology, we plan to enhance collaboration provenance management performance. Third, we plan to conduct more experiments to study the effects of tuning various parameters (e.g., the number of concurrent collaborators, the productivity of individual members, the number of tasks comprised in the shared scientific workflow) on concurrent productivity. Fourth, we plan to explore conducting collaborative scientific workflow composition in the Cloud infrastructure.

8. REFERENCES