Pipelining Agents for Language Services and NLP Components
Arif Bramantoro 1, Toru Ishida 2

Abstract: This paper proposes a pipelining agent architecture to integrate two different frameworks in natural language processing research, i.e. Heart of Gold and Language Grid. The agent architecture is chosen to automate the integration with two main tasks, creating processing flow and modifying workflow. Another reason of using agent architecture is that it supports the combination of pipelined language services in the Language Grid and the pipelined natural language processing components in Heart of Gold, with different treatment of each combination. The integration itself aims at providing more language resources available on the Web and contributing to interoperability among various language services.

Keywords: language services, NLP components, Pipelining agents

1. Introduction
To accelerate the rate of scientific discovery, computer scientists collaborating with computational linguists today have been trying to develop more and more different kinds of natural language processing (NLP) infrastructures. Two kinds of the famous ongoing developments of NLP infrastructures are based on Web services and software components. Service-based NLP infrastructure uses workflow to enable particular language service to be pipelined with other language services. Component-based NLP infrastructure uses processing flow to enable particular NLP component to be pipelined with other NLP components.

The heterogeneity of NLP infrastructures delivers a massive problem in pipelining more than one processing tool. We address this problem by proposing a Pipelining agent framework for two primary kinds of NLP infrastructures, language services and NLP components. This novel agent framework enables the automation of pipelining workflow in language services and processing flow in NLP components.

To investigate the usefulness of Pipelining agent framework, we use two language infrastructures for language services and components, the Language Grid [2] and Heart of Gold [8]. Our hypothesis is that a wide range of natural language application can be developed from this integration and agent framework.

The Language Grid is a service based collective intelligence framework which enable access to various language services in the world based on a single powerful protocol, HTTP. For the Language Grid, the more language services it has the better it is for the availability of composite services. Composite language service means the ability to create a new service by combining existing services by using service workflow.

Heart of Gold is also a framework that tries to accommodate any natural language processing (NLP) tools as NLP components regardless the depth of the analysis. The return value of this framework is an XML standoff annotation string which can be further processed by any application. Heart of Gold provides a high degree of flexibility for accessing its server as well as defining processing flow for integrating composite NLP components [7].

The Language Grid as a collective intelligence framework of pipelining services can be enhanced through the integration with any pipelining software, for example, Heart of Gold, by utilizing a software agent to pipeline different sources of language processing tools. Therefore there will be more language services available in the Language Grid environment. The integrated architecture should treat Heart of Gold not only as atomic service in the Language Grid, but is also able to pipeline other services in the Language Grid. We argue that by declaring the pipeline system as an agent platform, the integration between Heart of Gold and the Language Grid becomes more interoperable.

The remainder of this paper is organized as follows: Section 2 describes the proposed architecture of Pipelining agents. Section 3 details the two different infrastructures of natural language processing. Section 4 presents how the hybrid architecture is supported with pipelining agents. Finally, in Section 5, the main conclusions and future works are highlighted.

2. Pipelining Agents
We propose Pipelining agent and its architecture to pipeline different types of NLP infrastructures. Two types of NLP infrastructures are currently supported by this agent, component-based and service-based NLP infrastructures. The underlying agent architecture for Pipelining NLP tools as described in Fig. 1 consists of three kinds of agents, User agent, Processing Flow Creation agent and Workflow Modification agent. Three repositories are utilized by
these agents, i.e. NLP Component Information, Processing Flow and Workflow repository. We assume that each agent has static and dynamic knowledge, a set of acquaintances, and a mailbox. One agent communicates to another by exchanging point-to-point asynchronous messages.

2.1. User Agent
User agent acts as a bridge between the user and NLP system. This agent is required to create two other agents. It has a responsibility to check the necessity to call Processing Flow Creation and Workflow Modification agent online. Unless the user explicitly define his request composited with the need of a new combination of processing flow and workflow, the decision to other agents online is based on the frequency of how many times the language services and NLP components are added in the repositories. Another parameter to call other agents online is the availability of the composite language services and components. This is the main reason why we implement agent framework in pipelining language services and components, since agents are aware of their environment and can modify it, in essence creating processing flow online.

When a user requests a particular task to be performed by composite language services, the User agent calls Processing Flow Creation agent to create a new processing flow based on NLP component information in a repository, and then calls Workflow Modification agent to replace particular atomic service in the existing workflow with the one described in the processing flow if necessary. The requested workflow then can be delivered to user by the User agent.

2.2. Processing Flow Creation Agent
As its name suggested, the main task of Processing Flow Creation agent is to create the processing flow of composite NLP components. The creation is made based on the NLP component information stored in NLP Component Repository which is assumed to be already provided by linguistics. The most important information stored in this repository is NLP depth value. This depth value indicates the ability to process natural language sentence. The deeper NLP component, the more functionality it has.

One of the processing flows used in the natural language research is written in SDL [3]. SDL, stands for System Description Language, is a specific language initially used for building NLP systems and pipelining multiple NLP components. SDL uses a declarative specification language to define a flow of information (input and output) and a pipelining process between linguistic processing components. The declarative specification consists of operators, symbolic module names, assignment of these symbolic module names to Java class names, constructor arguments and some other processing options. The basic operators currently available in SDL are as follows.

- + for sequence. A component runs in sequential pipeline after the previous component has finished.
- | for parallelism. Multiple components run in parallel by using Java threads inside the pipeline.
- for unrestricted iteration. A component runs in a loop until its output remains unchanged.

A set of syntactically well-formed component descriptions \( D \) in SDL can be formulated based on an initial set \( C \) of (atomic) components as follows.

\[
\begin{align*}
C & \subseteq C \\
d1, d2 & \subseteq D \Rightarrow d1 + d2 \subseteq D \\
d1, \ldots, dk & \subseteq D \Rightarrow (d1, \ldots, dk) \subseteq D \\
d & \subseteq D \Rightarrow (*) \subseteq D
\end{align*}
\]

The Processing Flow Creation agent runs either online or offline. When all NLP component information has been accommodate in the repository, the online call is not needed.
2.3. Workflow Modification Agent
The main task of Workflow Modification agent is to modify the existing workflow of composite language services. A modification is made based on the depth value of NLP component. If a composite NLP component in the repository provides more functionality, i.e. deeper value, than particular atomic service in the workflow, the agent will replace it. If there is no additional functionality, the agent will not modify the existing workflow.

Composite services which are composed of multiple services are described in particular workflow language. One of the workflow language commonly used for language service is WS-BPEL [6]. While processing flow in SDL is possible to be written automatically, workflow in WS-BPEL is not possible. This service composition language is usually not written by hand, but drawn in graphical environment which is out of scope in this paper. We only concern on how to replace one atomic service in the workflow with a service provided by Heart of Gold, either atomic or composite NLP components.

The Workflow Modification agent runs either online or offline. Here online means that the agent can be merged on demand with other Web services during service request. An existing workflow is assumed to be well established in the Extended Workflow repository by language service designer. An alternative workflow is automatically created and stored in the Extended Workflow repository together with its generated SDL description of incorporated NLP components.

The Workflow modification agent could also run offline to optimize the creation of a new workflow before receiving a request from service user. This agent runs offline especially when language services and NLP components are added frequently. If the required task is also not available on the repository, the Workflow modification agent is required to produce a new combination of processing flow and workflow. The advantage of running the Workflow modification agent offline is that the processing time of a user request is not affected since the new workflow has already been stored in the repository before runtime.

If the Workflow Modification agent has no solution to add an SDL into the existing workflow, the agent will store nothing to the repository. This is usually due to the incompatibility between language services and components. For example, the output of NLP components contain less than what language services requires in its service input, and vice versa.

3. Two Platforms of Pipelining NLP Tools
Two of the ongoing developments of natural language infrastructures are the Language Grid and Heart of Gold. We took these two models of natural language infrastructure since they represents most infrastructures based on language services and components.

3.1. Language Grid: Pipelining Services
The Language Grid [2] is an infrastructure for coordinating distributed language services on the Web. Various services such as machine translators, morphological analyzers, bilingual dictionaries and parallel texts, concept dictionaries, and text-to-speech are available on the Language Grid.

The Language Grid has functionalities for access control to deal with intellectual rights issues of language services, and consists of two kinds of nodes; core nodes and service nodes. Core nodes take charge of a repository and management of services available as part of the Language Grid. Language services are deployed on service nodes by service provider and accessed from core nodes. A service user, which can be either a client application or collaboration tool provided by the Language Grid; first sends a request to a core node. The core node next finds the required services searching its repository or inquiring other core nodes. Then the core node invokes services found on service nodes. During this process, the core node authorizes the client in order to confirm the access rights and the number of invocation which the client is allowed. The language service provider can configure the permission and monitor the statistics of use of her resource anytime.

3.2. Heart of Gold: Pipelining Components
Heart of Gold is a middleware that bridges user application and external NLP functionalities. Heart of Gold provides an integrated pipeline between deep and shallow NLP annotations. Deep NLP applies as much linguistic knowledge as possible to analyze natural language sentences [1]. On the other hand, shallow NLP neglects the use of the whole range of linguistic details, but concentrates on specific aspects.

The main advantage of using Heart of Gold is the availability of deep NLP for computing deep semantics representations for natural language sentences (sentence semantics). For example, as similarly illustrated in [7], when we receive a sentence “Tom gave his son a toy” then we can have a deep analyzed result as past\(\text{give}(\text{Tom}, \text{his son}, \text{toy})\). This deep analysis can be used to create other syntactic variants such as “A toy was given by Tom to his son” or “Tom gave his son the toy”.

4. Pipelining Agents in Hybrid Architecture
In our previous works [4,5], two NLP infrastructures, the Language Grid and Heart of Gold, were combined by wrapping Heart of Gold as a Web service and register it in the Language Grid environment. Here, we propose to implement pipelining agent architecture to support the integration.

The architecture of the Language Grid consists of four layers, P2P Grid Infrastructure, Atomic Language Services, Composite Language Services, and Intercultural Collaboration Tools. We propose that the
Language Grid can utilize Heart of Gold by adding it to the Atomic Language Services layer. Although it is not common in the Language Grid to have composite NLP components in the layer of atomic language service, the standard wrapping technique of the Language Grid requires Heart of Gold to be located in this layer. Consequently, we have to treat differently Heart of Gold with the support of pipelining agents in this layer since it contains many NLP functions that behave as composite services.

Although Heart of Gold is integrated in the Language Grid as a service, the components orchestrated in Heart of Gold service cannot be considered as composite services. This is due to the wrapping technique of Heart of Gold in the Language Grid is simply putting the service in the Atomic Language Service layer of the Language Grid and, therefore, considered as an atomic service. Without considerable effort, it is not possible to split all components in Heart of Gold and wrap each component individually in the Language Grid. One reason is that all components in Heart of Gold are configured to be compatible to Heart of Gold’s middleware and not to others.

By utilizing Pipelining agents, the automatic composition between language services and NLP components could be realized. A new workflow integrated with its processing flow is added into the existing workflow based on service and components information. Thus, Heart of Gold could be accessed as composite service in the Language Grid environment. The role of Pipelining agent architecture is described in Fig. 2.

**Fig. 2.** The role of Pipelining agents in accessing Heart of Gold as composite service in the Language Grid environment

5. Conclusion

The use of agent platform for Web services and XML/RPC based framework gave us the opportunity of mixing different kinds of natural language processing tools in a common system, allowing us to take advantage of each tool’s strong points. In this paper, we have proposed a Pipelining agent to integrate two complex frameworks, Heart of Gold and the Language Grid. To pipeline NLP functions in Heart of Gold, we wrapped Heart of Gold into a Web service in the Language Grid environment. By pipelining Heart of Gold service in the Language Grid, we can exploit distributed language resources in NLP-based applications.

A future work is needed to examine the language resources with different characteristic that can be combined with the help of agent framework. The composite NLP components in Heart of Gold should be pipelined with the existing workflow of composite services in the Language Grid environment. On the other hand, the composite language services in the Language Grid should be pipelined in the processing flow of Heart of Gold components.

**Acknowledgments**

This research was partially supported by Strategic Information and Communications R&D Promotion Programme from the Ministry of Internal Affairs and Communications, and also from Global COE Program on Informatics Education and Research Center for Knowledge-Circulating Society.

**References**


