Planning for Automatic Video Processing using Ontology-Based Workflow

Gayathri Nadarajan
Artificial Intelligence Applications Institute (AIAI)
CISA, School of Informatics
University of Edinburgh, Scotland

Abstract

The provision of automatic workflow composition within the Grid is challenging and requires more attention. Planning, combined with semantic-based technologies such as ontologies could prove useful in composing automatic workflows for pervasive problem domains such as video processing. We outline a framework that incorporates these features and seek a Planning model that would contribute towards a semantically rich and performance-based workflow.

Introduction

Automatic workflow composition in dynamic environments such as the Grid (Foster & Kesselman 2003) is challenging because it is difficult to capture the functionality of each workflow component based on high level user requirements (Yu & Buyya 2006). Yet it would prove useful for large scale workflows which are very time consuming to compose manually, such as those for automating video processing tasks. These tasks are complex, repetitive and usually domain-specific in nature. Combining the flexible and generic characteristics of workflow systems could help make the highly specialized and often handcrafted vision problem-solving more modular.

To this end, major Grid workflow systems are composed manually, including Pegasus (Deelman *et al.* 2004), Triana (Taylor *et al.* 2003), Taverna (Oinn *et al.* 2004) and Kepler (Ludäscher *et al.* 2005). Thus the user, who is usually a domain expert (e.g. bioinformaticians using Taverna), will be responsible for constructing the workflow based on their goals. Only Pegasus has the additional capability of automatic workflow composition in the form of mapping abstract non-executable workflows to their concrete executable forms. Pegasus utilizes deferred planning to generate partial executable workflows based on already executed tasks and the currently available resources by a partitioner. This allows for dynamic scheduling that would prevent workflows from failing to execute should any of the resources fail. Although this is a step towards performance optimization and reliability, Pegasus is still limited in that it does not support looping which is essential for the modeling of iterative processes such as image processing.

The primary motivation for this work is triggered by another existing need – a vast amount of real-time videos from an ecological source in the EcoGrid (2006) that need to be analyzed effectively and efficiently. Continuous collection of data of varying qualities using wireless sensor nets within the EcoGrid in Taiwan has caused manual processing by human experts (ecologists) too time consuming. A suitable solution would require appropriate automated methods, performance-based selection, adaptive, flexible and generic architecture and semantically rich to enhance Grid-compatibility. We aim to provide an automatic workflow composition method utilizing Planning that would provide an alternative solution for video processing tasks which are traditionally conducted manually.

Proposed Framework

Based on the ecological challenges and technical motivations in the previous section, we proposed a semantic-based hybrid workflow composition method within a three-layered framework that distinguishes different levels of abstraction through the design, workflow and processing layers (Fig. 1).

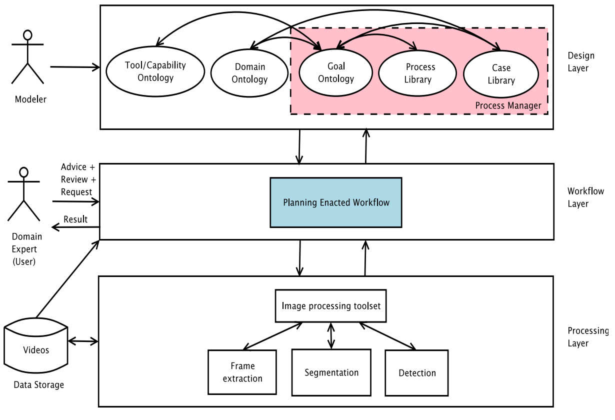

Figure 1: Hybrid Workflow Composition Framework for Video Analysis (Nadarajan, Chen-Burger, & Malone 2006a).

The central idea of this architecture is that users who do not possess image processing expertise can conduct complex

---

video processing tasks within a dynamic environment. This is realized via a Planning- and ontology-based workflow enactor that acts as the main interface between the high-level user requests and the low-level application components.

The design layer contains components that describe the goals, domain descriptions, capabilities and processes to be carried out in the system. These are represented using three ontologies (see next section) and two libraries. The process library holds instances of executable processes to perform the tasks of the workflow while the case library keeps track of previous viable solutions and finds a similar solution to match the current problem whenever there is more than one possible solution or no solution for a particular goal. A modeler is able to manipulate the components of this layer, for example populate the libraries and modify the ontologies.

The workflow layer acts as the main interface between the design and processing layers as well as the user. It ensures the smooth interaction between the components, access to and from various resources such as raw data, video and image processing toolset, as well as the provision of the final output to the user. A Planning- and ontology-enriched workflow enactor acts as the interpreter of the events that occur within the system and plays the important role of choreographing the flow of processing within the system.

The processing layer consists of a set of video and image processing tools that will act on the data. The functions of these tools are represented in the capability ontology in the design layer. Once a workflow has been established, these tools may work on the videos directly. The final result is passed back to the workflow layer for output and evaluation.

Linking the Ontologies with the Planner

In order to realize this framework, an initial development of the ontologies and a walkthrough for a detection example have been implemented manually (Nadarajan & Renouf 2007). We have opted to incorporate three ontologies to separate the goals from the capabilities and to provide meaning for the process within a semantically integrated system. Each ontology holds a vocabulary of classes of things that it represents and the relationships between them. The goal ontology contains the high-level goals and constraints that the user will communicate to the system, the domain ontology describes the concepts and relationships associated with the videos, such as the lighting conditions, color information, position, orientation as well as spatial and temporal aspects. The capability ontology contains the classes of video and image processing tools and their functionalities. The use of ontologies is beneficial because they provide a formal and explicit means to represent concepts, relationships and properties in a domain. A system with full ontological integration has several advantages; it allows for cross-checking between ontologies, addition of new concepts into the system and discovery of new knowledge within the system.

As a sample application, the user will provide the domain description along with the goal and constraints of the problem to the system. The user request and domain knowledge are captured via the goal and domain ontologies. For instance, a typical user request could be *“Detect all tiger fish in bright videos”*. Thus the goal is identified as ’Detection’, the constraint as ’Occurrences = All’ (both in goal ontology), and the domain descriptor as ’Lighting = High’ (domain ontology). In the goal ontology, each goal is linked to the high-level processes or sub-goals that are associated with it, such as ’Pre-processing’, ’Feature Extraction’, ’Segmentation’ and ’Classification’. The instances of these processes and their sub-processes are contained in the process library and will be selected based on task decomposition. The case library will contain past solutions with performance indicators attached to them to provide alternative solutions should none or more than one solution is found. Thus the role of Planning is to generate a sequence of actions which are represented by a set of primitive activities, or *capabilities*.

The capabilities and associated tools are captured within the capability ontology. A tool is a software component that can perform a video or image processing task independently, or a technique within an integrated vision library that may be invoked with given parameters. Once a set of tools is identified, the workflow for the initial high-level goal is composed and can be scheduled for execution in the processing layer.

An Example: Planning for ’Detection’ Task

Composing workflows for video processing tasks can be seen as a Planning problem where the goals are high-level user requirements, such as ’Detection’, ’Enhancement’, ’Reconstruction’ and so on. The operators are the tools or capabilities that are involved in achieving these goals. As will be illustrated below, a task-based Planning procedure would be suitable for solving video processing problems. Thus we have considered using Simple Task Network (STN) Planning as a starting point for our work.

For a simple detection task such as *“Detect all the fishes in the videos”*, the process model for this task is obtained. Fig. 2 outlines three high-level non-primitive tasks for achieving this goal.

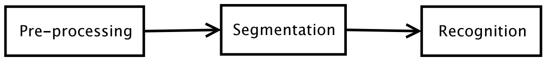

Figure 2: High-level process breakdown for ’Detection’.

Due to time and space limitations, we will explore the task breakdown for ’Segmentation’ only. One possible segmentation method is ’Background Subtraction’. This is represented in the process library as follows (shown in Prolog syntax):

```
segmentation(X) :- background_subtraction(X).
background_subtraction(_) :-
 background_model_construction,
 model_differencing, background_model_update.
```

’Background Model Construction’, ’Model Differencing’ and ’Background Model Update’ are each made up of several sub-processes, which in turn may be decomposable.

Using the conventions provided by Nau, Ghallab, & Traverso (2004), this could be described using `methods` for non-primitive tasks (Figs. 3 and 4). The parameters to the tasks and subtasks have been omitted for now.

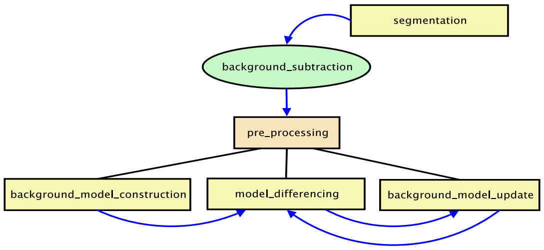

Figure 3: Decomposition for 'Background Subtraction'.

background_subtraction
task: segmentation
precond: pre_process
network: $u_1 =$ background_model_construction,
$u_2 =$ model_differencing,
$u_3 =$ background_model_update,
$\{(u_1, u_2), (u_2, u_3), (u_3, u_2)\}$

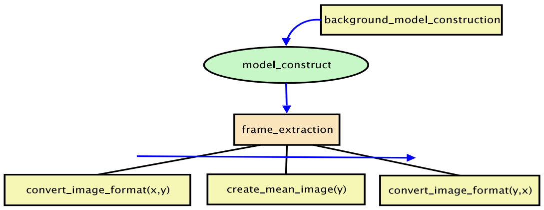

Figure 4: Decomposition for 'Background Model Construction'.

model_construction
task: background_model_construction
precond: frame_extraction
subtasks: <convert_image(x,y),
create_mean_image(y), convert_image(y,x)>

This simple illustration shows that Planning, in particular HTN Planning, is a suitable approach for composing workflows for video processing tasks. This is because there are many ways to achieve a vision task, depending on the quality of the video, the tools available to perform the task, and many other factors. HTN Planning would allow different methods for solving the same task (e.g. 'Detection', 'Segmentation') to be incorporated into the inference engine. To achieve optimum solutions, a performance indicator is also included as a parameter after a task has been performed using the set of tools determined by the Planner into the case library. Thus, by capturing some of the *best-practices* used by image processing experts to solve vision problems, we could incorporate this knowledge into the system.

At present we are working with image processing experts to perform various analyses on the EcoGrid videos using different image processing libraries in order to obtain the process models and performance levels of the different tools used. These tasks include detection, segmentation, classification and object tracking.

Conclusions

We have proposed the composition of a performance-based workflow which utilizes Planning and ontologies. Applying HTN Planning is advantageous as it can encode the heuristics that vision experts use to solve video processing tasks within the system. Thus the construction of the Planner is bottom-up because it is experience-based. To enable Grid-compatibility, the components in the proposed framework (Fig. 1) could be wrapped as Grid services (Nadarajan, Chen-Burger, & Malone 2006b). The techniques that we have discussed in this work will prove useful for automatic Grid workflow composition for pervasive problem domains such as video processing.

References

Deelman, E.; Blythe, J.; Gil, Y.; Kesselman, C.; G.Mehta; Patil, S.; Su, M.; Vahi, K.; and M.Livny. 2004. Pegasus: Mapping Scientific Workflows onto the Grid. In *Across Grids Conference 2004, Nicosia, Cyprus.*

EcoGrid. 2006. National Center for High Performance Computing (NCHC), Taiwan. http://ecogrid.nchc.org.tw/.

Foster, I., and Kesselman, C., eds. 2003. *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann, 2nd edition.

Ludäscher, B.; Altintas, I.; Berkley, C.; Higgins, D.; Jaeger, E.; Jones, M.; Lee, E.; Tao, J.; and Zhao, Y. 2005. Scientific Workflow Management and the Kepler System. *Concurrency and Computation: Practice & Experience.*

Nadarajan, G., and Renouf, A. 2007. A Modular Approach for Automating Video Analysis. In *12th International Conference on Computer Analysis of Images and Patterns (CAIP 2007), Vienna, Austria.*

Nadarajan, G.; Chen-Burger, Y.-H.; and Malone, J. 2006a. Semantic-Based Workflow Composition for Video Processing in the Grid. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006), H.K.* pp. 161–165.

Nadarajan, G.; Chen-Burger, Y.-H.; and Malone, J. 2006b. Semantic Grid Services for Video Analysis. In *Web Intelligence and International Agent Technology Workshops, (SERCOMP 2006).* pp. 121–124.

Nau, D.; Ghallab, M.; and Traverso, P. 2004. *Automated Planning: Theory & Practice*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Oinn, T.; Addis, M.; Ferris, J.; Marvin, D.; Senger, M.; Greenwood, M.; Carver, T.; Glover, K.; Pocock, M.; Wipat, A.; and P.Li. 2004. Taverna: A Tool for the Composition and Enactment of Bioinformatics Workflows. *Bioinformatics*. Vol. 20, No. 17, pp. 3045–3054.

Taylor, I.; Schutz, B.; Sathyaprakash, B. S.; Shields, M.; Wang, I.; Harrison, A.; Robinson, C.; Al-Shakarchi, E.; Kelley, I.; and Zhu, R. 2003. The Triana Project, Cardiff University. http://www/trianacode.org.

Yu, J., and Buyya, R. 2006. A Taxonomy of Workflow Systems for Grid Computing. *Journal of Grid Computing*. Volume 3, Numbers 3-4, pp. 171–200.