Semantics-Based Workflow Composition for Video Processing

Gaya Nadarajan

Jessica Chen-Burger
AIAI,CISA

Bob Fisher
IPAB
Introduction – Ecogrid

- Utilises Grid technologies to establish an infrastructure for ecological research in several locations in Taiwan, under the National Centre for High-Performance Computing (NCHC)

- Divided into four components; network, data streaming, data management and **workflow enactment (AIAI)**

Fig. 1 Sample images captured from EcoGrid video clips
Grid Workflow – Pegasus

- Planning for Execution in Grids (ISI, Uni. S. California)
- Workflow mapping engine (workflow instance → executable workflow)
- Abstract forms not bound to specific Grid resources while concrete forms are
- Concrete Workflow Generator (CWG) doesn’t perform a search, but makes a random choice whenever several alternatives are possible
- Uses Condor’s DAGMan for workflow execution and Globus to provide the middleware
- Doesn’t support loops, also doesn’t make use of ontologies which promotes reusability
Grid Workflow – Triana

• Open-source project developed at Cardiff University
• User constructs workflows graphically by drag-and-drop
• Supports custom (XML-like) and BPEL formats
• Supports looping and branching
• Requires user to construct workflows manually, also no integration of semantics to support interoperability over the Grid
Grid Workflow – Taverna

• EU project under myGrid initiative
• Users are biologists and bioinformaticians
• Makes use of semantic technologies, e.g. ontologies
• Workflow language used is Scufl and enacted using the Freefluo engine
• Supports implicit iteration mechanism and concurrency
• However, still requires user to construct workflows manually
Grid Workflow – Kepler

• Java-based, open source cross-project collaboration built on Ptolemy II based in UC Berkeley
• Mature application that allows scientists from several domains to design and execute workflows
• Uses an XML dialect called Modeling Markup Language (MoML) for workflows
• Highly reliable, supports looping and suitable for modelling complex tasks
• However, lacks automation in workflow composition
Knowledge-Based Vision – 80’s-90’s

• Expert systems were prominent, e.g. LLVE, CONNY, OCAPI, MVP and BORG

• They were motivated by (the then) sophisticated Fortran library and limited AI knowledge representation and reasoning methods

• Faced with many challenges, namely the lack of mature knowledge representation and reasoning techniques and the difficulty in generalising the image analysis process

• The former is now being addressed by Semantic Web technologies, e.g. ontologies
Knowledge-Based Vision – Orion

• Initiative by INRIA, France - addresses the problem of semantic image interpretation by providing a generic and reusable vision platform

• Used for the detection of plant diseases and for image indexing and retrieval purposes

• Visual concept ontology for object recognition

• Does not deal with the specification of the vision problem since the goal is always recognition
Knowledge-Based Vision – aceMedia

• EU-funded project managed by the Informatics and Telematics Institute, Greece

• Purpose is integrating knowledge, semantics and content for automatic video annotation achieved by the Autonomous Content Entity (ACE)

• Also aims to provide support for ontological modelling to allow for semantic-based analyses

• Incorporate several ontologies to describe the domain and image processing aspects but does not tie the goals with the processes involved in solving them

• They also use quantitative measures as opposed to qualitative measures for describing their entities
Motivation I – Ecological

• A vast amount of real-time videos from an ecological source is acquired via wireless sensor nets in the EcoGrid, unanalysed.

• Manual processing by ecologists is time consuming:

1 minute’s clip will take 15 minutes to analyse, thus 1 year’s recording will cost human experts 15 years’ effort x 3 under water cameras ≈ 45 years

• Impractical situation – more appropriate automation is required!
Motivation II – Technical (Workflow)

- Automatic Workflow Composition for dynamic environments such as the Grid is challenging
- Difficult to capture the functionality of each workflow component based on user requirements
- Major Grid Workflow systems are composed manually (Pegasus, Triana, Taverna, Kepler)
- Only Pegasus has additional capability of automatic workflow composition by mapping abstract workflows to their executable forms – but still limited
Motivation III – Technical (Vision)

• Most of image processing is done for very specific problem domains using highly specialised tools

• Potentially a combination of software tools could be utilised in an optimal way, but difficult to determine parameters such as threshold values, etc

• Knowledge-based vision efforts try to address this issue, some gaps still remain (BORG, Orion, aceMedia)

• Present knowledge-based vision efforts do not completely tackle the problem of the application context description (domain information) and the effects this may have on the images

• Aim to provide a modular approach for video analysis by the reuse of software tools and heuristics capture
Research Gaps

• Present Grid workflow systems lack automatic workflow composition which would prove useful for users who are domain specialists but not technical experts (e.g. ecologists who want to perform IP tasks)

• There is a lack of modularisation in the way image processing problems are specified and solved in knowledge-based vision systems

• I propose to overcome these limitations by using a workflow solution that utilises a hybrid planning method
Requirements

- Process Automation and Plan Creation for Execution
- Iterative Processing
- Performance-Based Selection
- Adaptive, Flexible and Generic Architecture
- Semantics-Based Compatibility
Proposed Approach

• Workflow management system to automate the steps taken in video analysis
• Workflow composition is done using Planning (to follow)
• Framework is enriched with ontologies (to follow)
Fig. 2 Hybrid Workflow Composition Framework for Automatic Video Analysis (Nadarajan, Chen-Burger & Malone. Web Intelligence ‘06)
Design Layer

• Process Manager is responsible for composing sequence of processes to be executed based on available tools. Associated with:
  - Planning component
  - CBR component

• Ontologies give meaning to process and keep goals separate from capabilities and domain description
  - Goal
  - Domain
  - Tool/Capability
Planning-enacted Workflow Layer

• Main interface between design and processing layers, and between the user and the system

• Planner responsible for composition of a set of actions

• Workflow responsible for the execution of these actions
Processing Layer

• Consists of a set of image processing tools

• A tool is a software component that can perform a primitive image processing task independently, or a technique within an integrated vision library that may be invoked with given parameters

• Functions of tools represented in capability ontology

• Once a workflow has been established, tools may work directly on videos

• Depending on quality of video and task at hand, each tool will work on varying level of accuracy
Work in progress - Ontologies

- Three comprehensive ontologies have been created in collaboration with Laboratoire GREYC (Nadarajan & Renouf ‘07)
- Ontologies evolve as new concepts are discovered without having to change implementation
Goal Ontology
Capability Ontology
Work in Progress - IP Tasks

• Collaboration with A. Renouf (GREYC) on *fish detection* task using Pandore

• Collaboration with C. Spampinato (Catania) on *video classification, fish detection* and *counting* (tracking) tasks using OpenCV

• On-going collaboration with C. Spampinato on *fish classification* using OpenCV, Matlab with possibility for parallelisation (MPI)
Work in progress – IP tasks results

Fig. 3. Left: Detection task using Pandore. Right: Video classification, detection and counting using OpenCV
Work in Progress – IP tasks cont.

- Obtained sequence and decomposition up to primitive task level for mentioned IP tasks
- Specific IP tasks are represented as process models, taking into account control constructs
- The different ways to solve IP (sub)-tasks are encoded as HTN methods
Bringing it all together - Planner

• Planner interacts with user to obtain request (goal)
• User may also provide domain information (e.g. brightness level) and constraints on the goal
• Planner then interacts with the ontologies to obtain implementation-specific values for the input obtained by the user via a data language (an extension of FBPML DL)
• Planner decomposes the high level goal into sub-tasks and seeks to find the operators that can be used to solve this problem
• Instances of the available operators are contained in the process library
Walkthrough

“Detect all fish in video”
Walkthrough

 Ontologies

 goal  domain  capability

 User

 • goal
 • constraints
 • domain info

 Planner

 Workflow

“Detect all fish in video”
Goal = Detection
Detail Level = all_occurences
Performance Criteria = real_time
Walkthrough

```
 Ontologies

- goal
- domain
- capability

User
- goal
- constraints
- domain info

1. goal
2. domain
3. capability

Planer

Workflow

```

“Detect all fish in video”
Goal = Detection
Detail Level = all_occurrences
Performance Criteria = real_time

detect = pre_process, segmentation, recognition
Walkthrough

Ontologies

- goal
- domain
- capability

User

- goal
- constraints
- domain info

Planner

- decompose $<o_1, o_2, ..., o_n>$

Workflow

Process library

"Detect all fish in video"

Goal = Detection

Detail Level = all_occurences

Performance Criteria = real_time

detect = pre_process, segmentation, recognition

Check for existence

$p1 = \text{convert_image}(x, y)$
Walkthrough

Ontologies

- goal
- domain
- capability

Process library

User

- goal
- constraints
- domain info

Planner

1. goal
2. domain
3. constraints
4. domain info
5. capability

Workflow

<o₁, o₂,...,oₙ>

“Detect all fish in video”

Goal = Detection
Detail Level = all_occurrences
Performance Criteria = real_time

detect = pre_process, segmentation, recognition

Check for existence
p1 = convert_image(x,y)

Check for performance
convert_image(x,y) = 3
Walkthrough

Ontologies
  goal
  domain
  capability

Process library
  detect = pre_process, segmentation, recognition
  Check for existence
  p1 = convert_image(x,y)
  Check for performance
  convert_image(x,y) = 3
  >1 solution/solution not found

Case library

User
  goal
  constraints
  domain info
  decompose
  <o_1, o_2, ..., o_n>

Planner

Workflow

“Detect all fish in video”
Goal = Detection
Detail Level = all_occurences
Performance Criteria = real_time

Goal = detect, detail = all_occurences, criteria = real_time, image_name, accuracy = 80%
Walkthrough

“Detect all fish in video”
Goal = Detection
Detail Level = all_occurences
Performance Criteria = real_time

detect = pre_process, segmentation, recognition
Check for existence
p1 = convert_image(x,y)
Check for performance
convert_image(x,y) = 3
>1 solution/solution not found

Goal = detect, detail = all_occurences, criteria = real_time, image_name, accuracy = 80%
<cvgCaptureProperty(), .., cvgReleaseCapture()>

Ontologies

goal
domain
capability

User

goal
constraints
domain info

decompose
<o1, o2,...,on>

Planner

Process library

Case library

Workflows

1. User inputs goal
2. Ontologies are used to understand the domain and capability
3. User inputs constraints and domain info
4. Planner receives inputs and decomposes into smaller tasks
5. Planner accesses Process library
6. Planner accesses Case library
7. Planner creates Workflow
Planner - initial ideas

Represent workflow composition as a planning problem, specifically using HTN

nodes = states, S, arcs = operators, O

Planning domain $D = (O,M)$

Planning problem $P = (s_i, w_i, O, M)$

Input: operators, methods, $M$

STN method, $m= (\text{name}(m), \text{task}(m), \text{precond}(m), \text{network}(m))$

HTN method, $m= (\text{name}(m), \text{task}(m), \text{subtasks}(m), \text{constr}(m))$

Plan, $\pi = \{a_1, \ldots, a_n\}$

$s_i = \text{initial state}, w_i = \text{initial task network}$
Planner cont.

Decomposition of task *Detection*:
STN methods

Some STN methods for Detection *(Nadarajan ICAPS DC’07)*

**background subtraction**

- task: segmentation
- precond: pre_process
- network:
  - u1 = background_model_construction,
  - u2 = model_differencing,
  - u3 = background_model_update,
  - \{(u1, u2), (u2, u3), (u3, u2)\}

**model_construction**

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

• Data language for representation of domain model (extension of FBPML DL to support multiple ontologies)
• Planner is Prolog-based
• Workflow layer, also Prolog-based with shell command to invoke IP programs
• Ontologies represented in OWL
Potential Contributions

• Automatic Workflow Composition
• A working system for a wider range of input for IP problems (flexible)
• Faster system construction for IP problems
Evaluation

Enhancement of workflow system will provide a working system with a wider range of input for Image Processing problems

• Flexibility
  
  • Carry out IP tasks on candidate systems with videos of varying qualities to test if they could perform the tasks on every data. Criteria for video qualities:

    i. Clear
    ii. Bright
    iii. Dark
    iv. Blur

  • Candidate systems include

    a. Manual, i.e. humans, e.g. domain and vision experts
    b. Existing Workflow Systems: Pegasus, Triana, Taverna
    c. My system
Discussion

• Development of planner – new vs. extend existing ones (JSHOP, Repair-SHOP, IX)

• Many components need to be integrated – focus on generic planner combined with ontologies and CBR
Schedule

- Thesis Writing
- Literature Review on Grid Workflow Systems
- Familiarise with Vision Tools and Techniques, Populate Initial Ontologies
- Manual Walkthrough of Fish Detection Task using Ontologies and Pandore Vision Library
- Background Reading on Planning and Application of HTN for Detection Task
- Collaboration with Vision Expert, Implementation of Planner
- Workflow Composition Tool – Integrate Planner with Libraries and Ontologies
- Evaluation

Date Range: Oct 05 to Jun 09
Thank you!

gaya.n@ed.ac.uk
References


• Nadarajan, G.; Chen-Burger, Y.-H.; and Malone, J. 2006a. Semantic-Based Workflow Composition for Video Processing in the Grid. WI 2006, Hong Kong.
