PhD Proposal:
Semantics-Based Workflow Composition for Video Processing

Gayathri Nadarajan (s0450937)
Supervisors:
Dr. Yun-Heh (Jessica) Chen-Burger
Prof. Robert B. Fisher

September 18, 2007

Abstract

We outline the problem of automatic video processing for the EcoGrid. This poses many challenges as there is a vast amount of raw data that needs to be analysed effectively and efficiently. Furthermore, ecological data are subject to environmental changes and are exception-prone, hence their qualities vary. As manual processing by humans can be time and labour intensive, video processing tools can go some way to addressing such problems since they are computationally fast. However, most video analyses that utilise a combination of these tools are still done manually. We propose a semantic-based hybrid workflow composition method that strives to provide automation to speed up this process. The requirements for such a system are presented, whereby we aim for a solution that best satisfies these requirements and that overcomes the limitations of existing Grid workflow systems that lack automation in their composition. This hybrid method uses Planning technology to decompose the video processing tasks, Case Based Reasoning to assist with performance-based selection and ontologies to provide contexts for the goals, domain description and IP tools available. Ideally, we aim to provide an approximate IP solution for naive image processing users, for instance video classification and fish detection in the ecological domain.

1 Introduction

1.1 Background

The EcoGrid project [3] is aimed at utilising state-of-the-art Grid technologies to establish a cyberinfrastructure for ecological research. This includes the integration of geographically distributed sensors, computing power and storage resources into a uniform and secure platform. Scientists can conduct data acquisition, data analysis and data sharing on this platform. The infrastructure is divided into four components; network, data streaming, data management and workflow enactment. The National Center for High-performance Computing (NCHC), Taiwan and Artificial Intelligence Applications Institute (AIAI), the University of Edinburgh have forged a research collaboration on workflow enactment in the EcoGrid [15].

Recognising the needs for real-time automatic and continuous information gathering through EcoGrid that offers a unique and immense opportunity for long term ecological monitoring and planning, NCHC has installed and managed Wireless Sensor Nets in several national parks in Taiwan. The information collected is stored in and made available through EcoGrid for access. The information collected includes surveillance videos in the Fu-Shan National Park covering the entire area for observing natural lives and protecting them from potential poachers, audio recording of frogs of rare species, under-sea coral reef and marine life observation stations and more. Due to the continuous and non-intrusive methods deployed, such monitoring and recording efforts have already made ecological discoveries of significant importance that traditional methods otherwise could not have made.

1.2 Ecological Motivation and Challenges

Continuous data collection in the EcoGrid, however, poses a great challenge as how this data may be transformed into usable information for the ecologists in a timely fashion. For instance, one minute of video clip is typically made up of 1829 frames and is stored in 3.72 Mbytes. That translates into 223.2 MB per minute, 5356.8MB per day and 1.86 Terabytes per year for one operational camera, and due to the unpredictability of nature, one may not easily skip frames as they may contain vital information. Based on our own experience, one minute's clip will on average cost 15 minutes' manual processing time. This means that one year's recording of a camera would cost human experts 15 years' effort just to perform basic analysing and classifying tasks. Currently there are three under water cameras in operation and this will cost a human expert 45 years just to do basic processing task. *This is clearly an impractical situation and more appropriate automation methods must be deployed.*

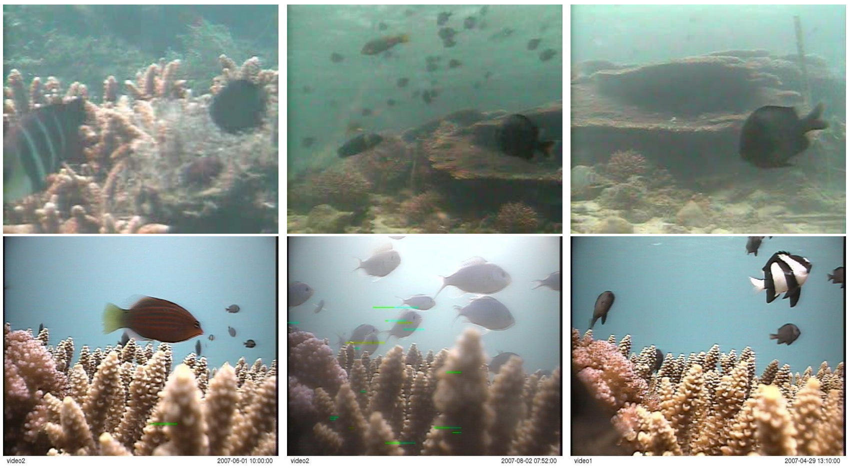


Figure 1: Sample images extracted from EcoGrid video clips.

The Grid [23] is an infrastructure for next-generation e-Science applications aimed at enabling resource sharing and coordinated problem-solving between computers and people in a distributed and heterogeneous manner. The Semantic Grid [46] is an extension of the current Grid in which information and services are given well-defined and explicitly represented meaning, better enabling this cooperation. This requires means for composing and executing complex workflows. Considerable research and implementation efforts have been made towards the development of workflow management systems for the Grid. Apart from the ecological challenge presented above, we are also investigating means to improve the technology of Grid workflow management systems by incorporating semantic capabilities and an improved automatic composition mechanism. By semantic capabilities we mean machine-processable knowledge and information, provided by enriched representations such as ontologies and Semantic Web-based languages [35, 26]. This would allow for a high degree of easy-to-use and seamless automation to facilitate flexible collaborations between researchers and virtual organisations, as imposed by the Semantic Grid community. Employing the power of distributed processing within the Grid will also help immensely with overcoming some of the ecological obstacles mentioned above.

2 Requirements

Given the introduction to the EcoGrid scenario and the motivation to tackle this problem which involves large-scale real-time data, a set of requirements that will correspond to the functionalities of a solution for a suitable system is presented in this section. These requirements are appropriate with the demands of ecologists who would be interested in analysing the videos and with our technical expertise in AI Planning and workflow composition. Sections 5, 6 and 7 describe the problem, proposed system and solution in more detail.

2.1 Process Automation and Plan Creation for Execution

A mechanism for automatically processing large amounts of data is required as this is the biggest problem that needs to be tackled and that it would be too time-consuming otherwise. In order to provide process automation, plans of predetermined actions (or processes) should be created. For example, the task "Annotation of Species X" could be decomposed into the following high-level steps that are required to be run automatically.

Figure 2: The High Level Steps Involved in a Video Annotation Process.

2.2 Iterative Processing

Each process in turn, could also be further decomposed into sub-processes. For instance, the "classification" process could be made up of a "segmentation" sub-process followed by a "recognition" sub-process. Segmentation could be made up of further sub-tasks. The recognition sub-process may also backtrack to the segmentation sub-process should it not succeed at the first few attempts. Thus, a mechanism that supports loops and task breakdown is required.

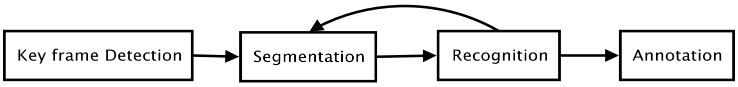


Figure 3: Process Decomposition and Iterative Processing for Video Annotation Example.

2.3 Performance-Based Selection

When a process can no longer be decomposed, it is scheduled for execution. For each of the final primitive processes, an associated software tool or technique is identified for it. It is possible that more than one tool or technique is available to perform this sub-task. The tool with the best capability should be selected to perform the desired function. Overall, the combination of tools with the *best performing capability* for the overall stated goal should be utilised. Thus the mechanism should be able to match the goal of the process with the capabilities (tools) based on a performance measure.

2.4 Adaptive, Flexible and Generic Architecture

Following from its definition, the Grid is a dynamic environment where availability of resources and their load can change notably from one time point to the next. An ideal system would be one that is extremely sensitive and adaptive to changing environments. It should enable distributed processing of tasks in different geographical locations. It would also possess a generic capability, whereby it should be able to perform a variety of video analysis tasks provided as input, for instance "recognition", "count", "compression" and so on. Furthermore, it should be able to generate new sequences of solutions based on past solutions. For this it should include techniques that allow it to incorporate its knowledge of previous experiences in order to generate new solutions as part of its learning.

2.5 Semantic-Based Compatibility

In addition to Grid and video processing requirements, the approach that we take should also be semantic-based, that is to integrate ontologies and Semantic Web-based languages, such as Web Ontology Language (OWL) [35], Temporal Resource Description Framework (Temporal RDF) [26] and more recently Web Services Modelling Ontology (WSMO) [45] to allow for effective reasoning in compliance with Semantic Grid requirements. An ontology-based system has several advantages such as the promotion of reusability of existing knowledge and the discovery of new knowledge within the system.

A workflow composition system would be ideal for automating repetitive tasks such as keyframe detection. Workflow systems are also well-known for managing and enhancing performance of processes [8], which ties in well with our requirement for a performance-based system. However, it remains a big challenge to provide a framework that encompasses all the requirements stated above. We now turn to existing workflow composition systems, and investigate how far they go in fulfilling these requirements.

3 Related Work

One key requirement of the Semantic Grid is facilitating the composition of multiple resources, and mechanisms for creating and enacting these in a distributed manner. This requires means for composing and executing complex workflows, which has attracted considerable effort. Here we provide a brief overview and analysis of several influential Grid workflow composition systems. Comprehensive studies on Grid workflow systems can be found in [51, 42].

3.1 Grid Workflow Systems

Pegasus. Pegasus [20] (Planning for Execution in Grids) is part of the GriPhyN project (Available as www.griphyn.org) and aims to support large-scale data management in a variety of applications such as astronomy, biology, gravitational wave-science and high-energy physics. It maps abstract workflows to their executable concrete forms. In an abstract workflow the workflow activities are independent of the Grid resources used to execute the activities. In a concrete workflow the workflow activities are bound to specific Grid resources and the necessary data movement to stage data in and out of the computations is included.

The abstract workflows can be user-defined or constructed automatically using Chimera [24], a Virtual Data System. Partial workflow descriptions that describe the input files, transformations (application components) and their parameters, as well as the output files produced by these transformations are fed to Chimera using the Chimera Virtual Data Language (VDL). Pegasus requires an input in the form of a DAG with XML description and produces a concrete (executable) workflow that can be given to the Condor's DAGMan [48, 4] meta-scheduler for execution.

Mapping the abstract workflow description to an executable form involves finding the resources that are available and can perform the computations, the data that is used in the workflow, and the necessary software through various Grid information services such as the Globus Replica Location Service (RLS), the Transformation Catalog (TC) and the Globus Monitoring and Discovery Service (MDS). When making resource assignment, Pegasus prefers to schedule the computation where the data already exist, otherwise it makes a random choice or uses a simple scheduling technique.

The concrete workflow is produced with a set of submit files necessary for its execution through DAGMan. These files are channeled as submit jobs to Condor-G, a component within Condor. DAGMan is responsible for enforcing the dependencies between the jobs defined in the concrete workflow. Pegasus utilises 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 optimisation and reliability, Pegasus is still limited in that it does not support looping which is essential for the modelling of iterative processes such as image processing.

Triana. Triana [49] is an open-source project developed at Cardiff University. It is a problem solving and workflow programming environment that allows users to construct workflows in a graphical manner. It has been used for text, speech and image processing tasks. A user creates a workflow by dragging the desired units from a toolbox and dropping them onto the workspace. Units are interconnected by dragging a cable between them. The resulting composite graph can be executed or saved. At present the Triana custom format (similar to XML) and BPEL4WS [12] are the supported formats. Workflows defined in these formats can also be read and handled by Triana. The GUI allows users to make changes to the workflow by adding, deleting or changing the sequence of execution by drag-and-drop. Additionally, Triana supports looping constructs, which is desirable from a process modelling point of view.

Triana is inherently flow based, it uses both data and control flow with component execution within Triana triggered by these flows. In the case of data flow, data arriving on the input "port" of the component triggers execution and in the case of control flow a control command triggers the execution of the component. Multiple inputs to a component can be set by the component designer to be mandatory, blocking until all received or optional, triggering immediately. The execution of workflow within Triana is decentralised, data or control flow "messages" being sent along communication "pipes" from sender to receiver. The communication can be either synchronous or asynchronous depending on the implementation of the communication pipe.

The internal workflow representation is object based, with specific Java objects for individual component instances or "tasks" and the hierarchy of connected tasks within a network. The representation is a Directed Cyclic Graph (DCG), cyclic connections are allowed within the Triana language, with nodes representing a component and vertices the connections between them. The external representation of the taskgraph is a simple XML syntax. A typical workflow consists of the individual participating component XML specifications and a list parent/child relationships representing the connections. Hierarchical groupings are allowed with sub-components consisting of a number of assembled components and connections.

Loops and execution branching in Triana are handled by specific components; there is a specific loop component that controls repeated execution over a sub-workflow and a logical component that controls workflow branching. The creators believe that this approach is both simpler and more flexible in that it allows for a finer grained degree of control over these constructs than can be achieved with a simple XML representation. Explicit support for constraint based loops, such as while or an optimisation loop, is often needed in scientific workflows but very difficult to represent. A more complicated programming language style representation would allow this but at the cost of ease of use considerations.

Control units within Triana dynamically rewire the workflow at run-time to connect to the remote services it has discovered based on the distribution policy and the number of services available. A middleware independent abstraction layer, called the Grid Application Prototype (GAP), enables Triana developers to advertise, discover and communicate with Web and P2P Triana Services. The GAP is used to interface with Triana services and provides the middleware independent view of the underlying services and interactions across the Grid.

Taverna. Taverna [41] is a collaboration between several European academe and industries under the myGrid project [47]. It aims to provide a language and software tools to facilitate easy use of workflow and distributed compute technology for biologists and bioinformaticians with limited technical background. The workflows are written in the Simplified conceptual workflow language (Scufl) and enacted using the Freefluo workflow enactment engine (Available at http://freefluo.sourceforge.net). It provides graphical interfaces that allows workflow manipulation and workflow progress invocation easily. As it is developed under the myGrid initiative, it makes use of semantic technologies to provide service descriptions that are closer to scientists' view of their experiments than implementation-specific syntactic types.

The Scufl language is data centric and is represented using a Workflow Object Model internally to Taverna. Within the execution flow layer, the complexity of the workflow design which is implicit within Scufl is interpreted by a lightweight data model that encodes the data that pass through a workflow. This data model contains some basic data structures such as lists and trees, which bring about an added complexity. Taverna uses an implicit, but configurable, iteration mechanism to handle this. It supports fault tolerance through a configurable mechanism; processors will retry a failed service invocation a number of times, often with increasing delays between retry attempts before, finally, reporting failure.

To provide a more flexible mechanism, Taverna does not attempt to formally structure the domain data. Taverna is particularly suitable for tasks that can handle simultaneous processing as it supports concurrency. However, Taverna does not support automatic composition of workflows.

Kepler. Kepler [31] is a visual, community-driven project with an extendable open source platform built on Ptolemy II [30], a mature application from the electrical engineering domain that allows scientists from several different domains to design and execute scientific workflows. It consists of a set of Java packages supporting heterogeneous, concurrent modelling, design, and execution.

Like Taverna, Kepler is dataflow oriented, with the core description being the processing of data through a set of connected actors (processing steps), thus it contains precisely defined models of computation. Kepler's strengths include its mature library of actors, which are mainly local applications, and its suite of directors that provide flexible control strategies for the composition of actors. It is also a modular, activity oriented programming environment that lends itself to the design of reusable components. These components, or processing steps include signal processing, statistical operations, and Boolean logic operations. Kepler can execute processes locally either within the Kepler environment (Java) or within a native environment (compiled native code, or code interpreted by another environment such as Perl).

The Kepler system models a workflow as a composition of independent components that communicate through well-defined interfaces. Workflows within Kepler are serialised in an XML dialect called Modeling Markup Language (MoML). Kepler performs both design-time and run-time type checking on the workflow and data. Processes can be executed in a distributed way, using Web and Grid services. Remotely executed processes behave as a single step in the model of computation regardless of their complexity. Kepler is based on a modular design where different execution models can be easily plugged into the workflows without changing any other components within the workflows.

Kepler supports looping and also has good reliability as it is able to produce partial results even when an entire workflow fails.

To a lesser extent the following Grid workflow systems were also investigated to provide a broader perspective.

Askalon. The Askalon [22] tool set is aimed at enabling performance-oriented, complex functionalities of parallel and distributed applications across the Grid. This is achieved through the provision of several sophisticated tools; the structure of each tool is based on the composition and sharing of remote Grid services, thus enabling tool interoperability. The Askalon workflow composition tool is visual and uses UML-based modelling and XML-based Abstract Grid Workflow Language (AGWL). Askalon has been designed as a distributed Grid service-based architecture and implemented on top of the Web services technology and Globus toolkit.

JOpera. JOpera [14] is a workflow tool for building distributed applications made out of heterogeneous parts. Its goal is to provide the composition of Grid services that is independent from the actual mechanisms and protocols and to provide a simple workflow language to do so. It uses a graph-based, functional, visual workflow modelling language embedded with XML and is available for Eclipse. The workflow lifecycle involves the selection of component services from a library, process building using visual drag, drop and connect facilities, running, testing and debugging within the same visual environment, execution and publishing of workflows as Web or Grid services. The JOpera workflow language uses data flow as the primary representation and explicit control flow for execution. It supports modularity, nesting and recursion. JOpera is quite versatile in that it supports various other service types including Web servers, SQL queries, Unix commands, Secure Shell (SSH), XML transformations, Java snippets and methods.

GrADS. GrADS [9] (Grid Application Development Software) is mainly concerned with improving workflow scheduling for Grid applications. It investigates various scheduling and rescheduling strategies and has been tested on a bio-imaging application. The heuristics employed within GrADS include min-min, max-min and suffrage heuristics. It builds an architecture-independent model for its workflow and employs analytical models that are constructed semi-automatically from empirical models in order to estimate the performance of a workflow component on a single Grid node. However it is not available for public use yet.

ICENI. [34] (Imperial College e-Science Network Infrastructure) provides component-based Grid middleware. Users construct an abstract workflow, which is a collection of components, using an XML based language that is then submitted for execution. The workflow language within ICENI includes all basic workflow structures, such as sequence, parallelism, choice and iteration. Starting from a spatial view, a temporal view of the workflow is derived showing the temporal dependence between the different workflow nodes.

3.2 Computer Vision Tools and Efforts

Many open source and commercial applications are available for performing a range of video processing tasks. These can be broadly categorised into libraries that provide a range of functions for low-level processing by vision developers, and also more integrated approaches that incorporate high-level knowledge and understanding from several fields for video analysis. However, there is a need to "bridge the semantic gap" between the easily-computable low-level content-based features and the high-level concepts which will be intuitive to the user [28]. Several tools and efforts of interest are outlined below.

3.2.1 Libraries

Intel's **OpenCV** (Open Source Computer Vision Library) [5] is a C/C++ based library of programming functions mainly aimed at real time computer vision. Example applications of the OpenCV library are Human-Computer Interaction (HCI), Object Identification, Segmentation and Recognition, Face Recognition, Gesture Recognition, Motion Tracking, Ego Motion, Motion Understanding, Structure From Motion (SFM) and Mobile Robotics. It includes built-in image processing functions such as filtering, linear transformations, geometric transforms, morphological erosion, statistics, image manipulations, thresholding and tiles image processing.

MATLAB® [1] is a well known and widely used commercial application by MathWorks. It is a high-level language and interactive environment that enables computationally intensive tasks such as matrix manipulation to be performed faster than with traditional programming languages and is suitable for a range of applications, including image processing. The MATLAB **Image Processing Toolbox** provides a comprehensive set of reference-standard algorithms and graphical tools for image processing, analysis, visualisation, and algorithm development. Its key features include techniques for image enhancement (linear and nonlinear filtering, filter design, deblurring, and automatic contrast enhancement), image analysis, colour image processing techniques, spatial transformations and image transforms.

Pandore [18] is an open source C++ based library developed at the GREYC Laboratory, France. Any operator from this library is a program performing an operation that cannot be further decomposed. Thus, any complex operation has to be decomposed into a sequence of several operators. Each operator takes as inputs images and numerical parameters and produces as outputs images and other (non-image) results. It can perform a specific operation on various image formats (pixel image, label image, region map, ...) and inputs / outputs are normalised. This library is particularly suitable for our research problem as each operator can represent a primitive activity or task in our process models.

3.2.2 Knowledge-Based Vision Approaches

Several notable efforts have contributed to providing knowledge assisted systems and frameworks for automating the task of video analysis. Since the 80's **expert systems** such as LLVE [33], CONNY [29], OCAPI [17], MVP [16] and BORG

[19] played a crucial role in providing knowledge-assisted video and image analysis. Automated image processing attempts were made by expert systems which generated executable programs from abstract commands. The development of such complex image analysis programs were motivated by (the then) sophisticated Fortran library and limited AI knowledge representation and reasoning methods. Expert systems were 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 such as ontologies, the latter is still a challenging task for computer vision. Here we present two recent knowledge-based vision projects.

The **Orion Project** [6] addresses the problem of semantic image interpretation by providing a generic and reusable vision platform utilising cognitive faculties. The aspects of cognitive vision involved are reasoning, learning and image processing mechanisms as well as knowledge ontology-based representation techniques. They propose a distributed architecture based on three highly specialised modules; a semantic interpretation module, a visual data management module and an image processing module. The platform is used for the detection of plant diseases and for image indexing and retrieval purposes. Two specific works from this project that will be discussed in the next section are [32] and [27].

Another project **aceMedia** [2] is one of the many EU-funded projects managed by the Informatics and Telematics Institute, Greece for automatic video annotation. Its purpose is integrating knowledge, semantics and content for user centred intelligent media services. The main concept introduced is the Autonomous Content Entity (ACE), which has three layers: content, its associated metadata, and an intelligence layer consisting of distributed functions that enable the content to instantiate itself according to its context (e.g. network, user terminal, user preferences). An ACE combines content, metadata, and intelligence into a single entity, allowing automated, knowledge assisted content processing. aceMedia is also aiming to provide support for ontological modelling to allow for semantic-based analyses. Although well-accepted and seems to be a promising effort, this project is still under development whereby a full working system is not available yet.

4 Research Gaps and Motivation

4.1 Limitations of Current Grid Workflow Solutions

The systems mentioned in section 3.1 possess some similarities and differences that are worth investigating in order to assess their suitability and limitations for our framework.

In terms of composition itself, Pegasus differs from Triana, Taverna and Kepler because instead of its abilities to compose and execute the workflows directly, it maps abstract workflows to their concrete forms, which are then executed by a scheduler. It also provides adaptivity through a partitioner that uses planning to produce partial executable workflows. The role of planning is important for our system as it allows for dynamic process selection and composition based on a given set of goals.

Triana, Taverna and Kepler provide a basis for users to create and run scientific workflows. Thus they strive to provide intuitive graphical user interfaces. These features could be adopted by our system as providing a user-friendly environment is essential to allow for ease of use. Both Kepler and Triana provide an interactive visual workflow editor while Taverna only has a static workflow viewer through its custom Scufl workbench [41]. Triana's interface is one of the most intuitive through its Visual Grid Application Toolkit (GAT), which could be benefited by Taverna to support interactive visualisation, but it isn't clear how this could be implemented [50]. While we believe that GUI development is important for our system, we are content to observe the usage of intuitive approaches provided by existing systems and expect to utilise these approaches in our system.

Triana, Taverna and Kepler provide a basis for users to create and run scientific workflows. All three systems contain similar elements; Triana's tasks are conceptually the same as Taverna's processes and Kepler's actors. The approach in Kepler is very similar to Triana in that the workflow is visually constructed from actors (java components), which can either be local processes or can invoke remote services such as Web services. In addition, our framework will incorporate elements to distinguish goals from the capabilities associated with the processes.

In terms of applicability, Pegasus would best suit a domain with well-defined requirements and where the overall goal could be determined from a given set of rules and constraints. Triana is well-suited for composing complex workflows for Web services and Peer to Peer services. Taverna is also suitable to be used in Web and Grid services contexts, but its use may be limited to composing simple workflows, whereas Kepler works very well for composing workflows for complex tasks but it has yet to reach its potential as a fully Grid-enhanced system since the system it is built upon (Ptolemy II) is primarily aimed at modelling concurrent systems. Furthermore, it is designed to be used by scientists which imposes some level of expertise to the user.

Incorporation of Semantic Web technologies within present systems are still limited. The use of such technologies should not be exclusively independent, rather they should be fully integrated into the system. Existing systems do not provide full ontological handling nor integration, instead they make use of separate ontology tools to define and manipulate ontologies. We wish to integrate the capability of reading ontologies (e.g. OWL files) and providing results and information in such formats as well.

A brief comparison of the four systems indicates that none of them could single-handedly fulfill all the requirements sought. For performing the complex task of video processing, Kepler stands out as a promising system to be considered as an underlying workflow composition mechanism. We wish to take advantage of the most prominent and advantageous features exhibited by existing systems, as well as incorporate new features that will fill the remaining gaps.

4.2 Gaps in Knowledge-Based Vision Systems

Most vision-based efforts concentrate on providing highly specialised techniques for very specific application domains due to the high demands for performance and accuracy in such problem domains. Many image processing experts design and develop applications from scratch each time, using trial-and-error cycles and not reusing already developed solutions [44, 43].

Earlier knowledge-based vision efforts cited in Section 3.2.2 (LLVE, CONNY, OCAPI, MVP and BORG) were limited to a list of restricted and well known goals. Therefore a priori knowledge on the application context (domain-specific concepts such as sensor type, noise, lighting, etc) and on the goal to achieve were implicitly encoded in the knowledge base. This implicit knowledge restricts the range of application domains for these systems and it is one of the reasons for their failure [21].

More recent approaches bring more explicit modelling but they are all restricted to the modelling of business objects description for tasks such as detection, segmentation, image retrieval, image annotation or recognition applications. They use ontologies that provide the concepts needed for this description, such as a visual concept ontology for object recognition within Orion [32, 27], a visual descriptor ontology for semantic annotation of images and videos within aceMedia [10] or image processing primitives. Alternatively they capture the business knowledge through meetings with the specialists, for instance the use of the NIAM/ORM method in [11] to collect and map the business knowledge to the vision knowledge. However, they do not completely tackle the problem of the application context description (just briefly in [32, 27]) and the effect of this context on the images (environment, lighting, sensor, image format). Moreover they do not define the means to describe the image content when objects are a priori unknown or unusable, for instance in robotics, image retrieval or restoration applications. They also assume that the objectives are well known (to detect, to extract or to recognise an object with a restricted set of constraints) and therefore they do not address their specification.

Thus it could be concluded that there is a lack of modularisation in the way image processing problems are specified and solved within knowledge-based vision systems. The Orion project does not deal with the specification of the vision problem since the goal is always recognition, while some work in aceMedia 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. We will illustrate how modularisation and qualitative descriptions are achieved by the use of several interrelated ontologies in Section 6.1.2.

5 Proposed Approach for Research Problem

At present it is clear that more attention should be dedicated to providing a more flexible framework that is context aware for the computation of complex tasks such as video processing in the Grid. Much work in vision systems, on the other hand, has been dedicated to providing specialised systems with high performance through the development of sophisticated and efficient algorithms via trial and error cycles. Such systems, however, are very costly as they utilise many man hours over many years. The ecological motivation posed by the EcoGrid and more recently, the case of finding a missing computer scientist (Jim Gray) using satellite images, have sparked an urgent need for a system that can do more generic automated video and image analyses in a short period based on approximate techniques. Most vision libraries, although comprehensive and are readily available, are not Grid-compatible.

Several research questions that need to be addressed have been identified. The research problems within the field of video and image processing are very hard and currently only specialised IP solutions are available to solve them. We attempt to use a rich and context-based workflow management solution which aims to provide a more flexible and generic approach to tackle this problem. We would like to investigate the advantages of our approach. Firstly, by striving to provide automation for video processing tasks using workflows, we aim for an alternative solution to existing manual approaches. Workflow management systems, planning systems and semantic-based approaches such as ontologies are generic by nature and we wish to extend this flexibility to video processing problems which are narrow. We also wish to investigate if incorporating Case-Based Reasoning with Planning could help enhance performance and the flexibility. As a side effect, by adopting these methods, we wish to investigate the possibility of reusing IP applications which are almost always developed from scratch each time a vision problem is tackled.

Summarising the main points from above, we propose a hypothesis for our research problem:
"The problem of automatic workflow composition for video processing tasks can be tackled by the use of a performance-based Planning approach assisted by Case-Based Reasoning and ontologies. These techniques may be used in a distributed computing environment, such as the Grid infrastructure. The solution plan could take into account performance measures such as execution time and quality of IP results for different algorithms used for solving the same task, different parameter values used in a module and the use of single versus distributed processors. The plans developed are tested on data from an ecological source of varying qualities and environmental conditions. Analysing such data is challenging due to the inherent uncertainties in the user requirements, suitable description for the data and the IP tools required to conduct the analyses. A context-based approach is useful to overcome these uncertainties by capturing known user requirements (goals), domain and IP software knowledge in ontologies, which are referred to by the planning-enacted workflow management system which could extract the relevant information required in order to find the solution to perform a task. Thus, this approach would allow us to rapidly develop a system which would normally cost domain and IP experts much effort and time to produce. This contributes towards the development of an approximate IP software that would allow domain or non-IP experts to perform complex video processing tasks much faster than they would manually."

6 Proposed Framework - Hybrid Method

We propose 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. The architecture diagram for this framework is given in Fig. 4.

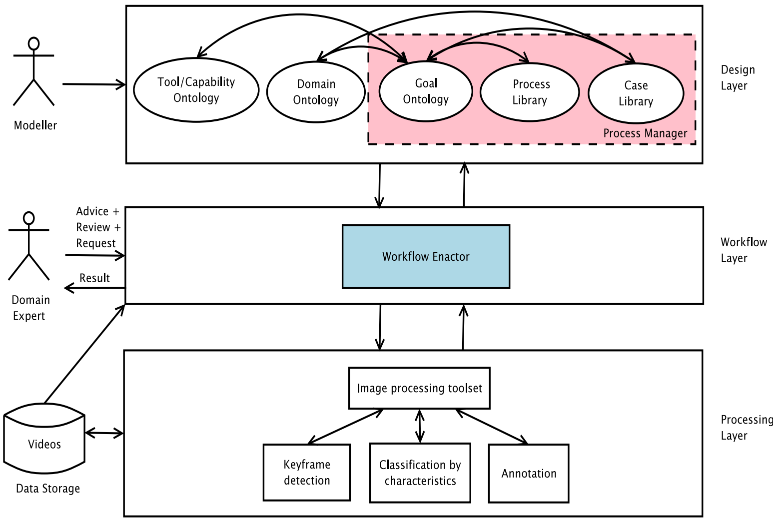


Figure 4: Overview of Hybrid Workflow Composition Framework for Video Annotation [37].

6.1 Design Layer

The design layer contains components that describe the domain, goals, capabilities and processes to be carried out in the system. These are represented using ontologies, elaborated in section 6.1.2, and two libraries. A modeller is able to manipulate the components of the design layer, for example populate the libraries and modify the ontologies.

6.1.1 Process Manager

The process manager is responsible for selecting and composing the sequence of processes to be executed in the workflow based on the tools available to perform the task. It has a Planning component, which comprises the goal ontology and the process library, and a Case-Based Reasoning (CBR) component, which is the case library. The process library holds instances of executable processes to perform the tasks of the workflow. Once a set of tools has been identified for achieving the goal, the process library composes the process sequence for execution. The goal ontology and process library, together, constitute the Planning component of the process manager.

However, there could be more than one possible solution or no solution for satisfying a particular task. While the Planning component is responsible for matching the goal with the capabilities (tools) and selecting the procedural steps to be taken, the CBR component is responsible for finding the closest solution from past scenarios for cases where Planning alone doesn't work. The case library keeps track of previous viable solutions and finds a *similar* solution to match the current problem. The heuristic for the closest solution is based on a similarity measure such as a modified form of the cosine matching function [25].

The instances of the processes in the case library will be associated with a workflow Quality of Service (QoS) measure. This could be a combination of one or more of *time, cost, reliability, fidelity* and *security* constraints [13]. This QoS "score" will determine which sequence of processes will constitute the best-performing option for the workflow composition.

6.1.2 Ontologies

We have opted to incorporate three ontologies to keep the goals separate 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; the goal ontology contains the classes of tasks (e.g. "Detection") with the qualifiers (constraints) for the goal. The domain ontology describes the videos, qualitative concepts such as "blur", "clear", "bright" are included. 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. They play an important role in fulfilling semantic interoperability, as highlighted in section 4.1. A workflow system with full ontological integration has several advantages. It allows for cross-checking between ontologies, addition of new concepts into the workflow system and discovery of new knowledge within the system.

6.2 Workflow Layer

This layer acts as the main interface between the design and processing layers. 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 workflow enactor acts as the interpreter of the events that occur within the system.

6.2.1 Workflow Enactor

The workflow enactor plays the important role of choreographing the flow of processing within the system. For the task of annotation, the flow of events could be summarised as follows. The user inputs a request into the system. The workflow enactor fetches data in the form of a video clip from the data storage and delegates the user request to the design layer. Based on the Planning and CBR approaches, a workflow for annotation will be established and passed to the workflow enactor. This is then fed to the processing layer for further action. The processing layer will communicate the annotated video back to the workflow enactor, which will pass this result to a domain expert who will provide feedback. This feedback mechanism could be used as a basis for improving the performance of the system. Finally, the case library is updated with this solution. A walkthrough on this flow of interaction is demonstrated in section 10.2.

6.3 Processing Layer

The processing layer consists of a set of video and image processing tools (section 6.3.1) 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.

6.3.1 Image Processing Toolset

A set of image processing tools and operators for performing various tasks would be available. Their functionalities are represented semantically in the capability ontology. It should be noted that there could be more than one tool available for each function, for instance, extracting the frames of a video clip could be performed by `mplayer` or `ffmpeg`, among others. We have also opted to make use of extensive open source libraries that could provide a range of image processing functions, such as OpenCV and Pandore (See Section 3.2.1).

6.4 Example Scenario [38]

The video clips from EcoGrid are characterised by various features that have to be processed by the system. These features include, for example, various species of fish, varying levels of brightness (e.g. sun light variation), complex and cluttered backgrounds and differing noise levels. The data is also captured from distributed sensors, thus the processing is to be conducted in Edinburgh for data that originated from Taiwan. Furthermore, users could be based in different locations, for instance, marine biologists from Taiwan, or their contacts in other places. Wrapping each tool in a service would enable each of them to be a Grid service. The workflow enactor and ontologies are also wrapped as services. Using Grid services, coupled with Planning and Case Based Reasoning (CBR), would enable a virtual environment to be built on top of a Grid infrastructure. This would allow distributed video analysis to be performed with more success than current technology could afford to perform. For instance, sensor grid nodes in Fu-Shan National Park capture data for a week that is required for our initial study. The user, a domain expert who wishes to classify a type of fish in a video, requests this by supplying the appropriate input parameters (i.e. goal and constraints). The data is packaged for Grid use and sent to our system where it is determined that tools “X”, “Y” and “Z”, each not necessarily from the same location, should be used on the video clip because the features best match these tools’ capabilities. The analysis is performed and sent back to the user. If the user is a domain expert, s/he could validate the accuracy of the classification performed by providing appropriate feedback to the system.

7 Methodology

7.1 Design

The components in the framework proposed are implemented in stages. Firstly the ontologies are populated; see section 10.1 for the three ontologies that have been created. Next the process library is created to contain instances of processes that are executable. See section 10.2 for an example of how this could be represented. The case library will be empty initially but as solutions are encountered with performance indication for given values, the case library will be populated accordingly. Thus the knowledge of the system can be increased incrementally. Next the Planner is implemented, this will be the main component to be developed. It will take in a goal and a set of constraints (if any) and automatically generate a plan using task decomposition. Thus the heuristics used by IP experts who have conducted video analysis will be incorporated into the Planner. Hierarchical Task Networks (HTN) Planning [40] is taken as a starting point to decompose the tasks [36].

Next the workflow enactor is implemented, as the workflow is composed automatically, there will not be a need for the user to compose the components of the workflow. However, it will take in a goal and constraints from the user. From that it will link the Planner with the ontologies, process and case libraries. For now, we assume that the parameters into the modules called within the vision libraries have set values, e.g. threshold values. However, it would be useful to incorporate a mechanism that would allow these parameters to be adjusted so as to generate an optimal solution.

The enactor is also interfaced to the image processing toolset, which are acquired/included in the following ways:

1. By using independent image processing tools identified that could perform specific tasks, e.g. ffmpeg, mplayer.
2. By using existing image processing techniques available in an extensive library, e.g. OpenCV, Pandore.

We anticipate that most IP functions will be acquired using the second method initially, while more independent software tools would be sought over time. Once all the components are created and interfaced with the workflow enactor, the system should be able to interface with human experts for feedback on accuracy. The use of a GUI would be suitable for this purpose.

7.2 Evaluation

Based on the proposed hypothesis in Section 9 and available test case from EcoGrid, the system developed is evaluated using the following criteria

1. Performance
   The test for performance is conducted by carrying out one or more video processing tasks on a sample test data (i.e. video clip) using our system and then the same tasks are repeated on the same data using other existing systems to determine which performs with optimal accuracy and speed. An example video processing task is outlined in Section 10.2. Candidate systems to be considered are as follows:

   - Manual, i.e. humans, e.g. domain experts (ecologists) and vision experts.
   - Existing Workflow Systems: Pegasus, Triana, Taverna.
   - My System.

   This test should result in smaller execution times for my system as compared to humans or other workflow composition systems.

2. Flexibility
   Testing for flexibility involves running video analysis on the candidate systems using videos of varying qualities to test which could perform the most tasks. Three different qualities of videos identified are:

   - Clear
   - Bright
   - Dark
   - Blur

   The suggested experiments include the tasks of fish detection, fish counting, and classifying the fish types for each type of video using my system and other existing workflow systems. My system should be able to process videos of varying qualities with a higher level of accuracy than existing systems.

8 Discussion

We have proposed a generic framework that enables an adaptive workflow enactment for video processing in the EcoGrid. This framework is based on a self-learned workflow composition method that utilises a hybrid approach of AI Planning and Case-Based Reasoning. We believe that full automation will be successful if the workflow tool has a full understanding of video processing tasks, but this is impossible for machine-implemented systems. Humans could provide feedback on some tasks to help improve this situation and enhance the performance of the system.

Implementation issues will need to be addressed for the proposed framework. At present, we have installed and used several open source products, mainly OpenCV and Pandore for image processing. We are also using Sicstus Prolog which is licensed within Edinburgh University for implementing the Planner. Should we want to provide a GUI for the Planner then we would consider using a visual-based Prolog software, at present the best tools are commercial. Thus the interface will be simple and text-based for now.

In discussing the approach taken by our framework, it's appropriate to consider process-oriented and data-oriented paradigms for video processing. The former treats processing (how things are done) as the first class primitive over data while the latter treats data (what is being manipulated) as the first class primitive over processing. Our framework, by utilising planning, supports the process-oriented paradigm, as opposed to most existing workflow composition systems. Prioritising processing over data, however, does not imply that the data is not of concern. By focusing on the techniques applied to the data, one is also implicitly giving importance to the data because the methods manipulate the characteristics of the videos and extract useful information from them. Thus the process-oriented approach complements the data-oriented approach.

A similar approach to the process-oriented paradigm is the state-based paradigm, where the next step to be executed is given by a set of allowed states from the current state based on predefined conditions. Although this method is useful for distributed environments, such as agent-based systems, its capability is limited to performing simple tasks, as defining states for complex tasks such as video processing would lead to a big state space.

We expect our work to contribute to strengthening video processing for workflows and vice-versa. Active and notable efforts in the development of Grid workflow systems could benefit from the semantic- and performance-based emphases that our framework proposes.

The main aim of this work is to provide a framework for automatic video analysis in the Grid utilising workflow technology. As this encompasses several fields of active research including Grid workflows, video and image processing, ontological engineering and machine learning, main contribution areas have been identified to emphasise the focus of this work.

9 Potential Contributions

Two main contribution areas have been identified for the scope of this work. They are outlined below and are in accordance with the proposed hypothesis (Section 5).

9.1 Workflow Inference

The main work is focused on the composition and reasoning mechanism provided by the workflow engine. The composition can be seen as the dynamic scheduling of the processes or activities in the workflow. Each primitive process can be executed by a tool which corresponds to a module within a vision library. An optimal resource allocation strategy should be adopted to ensure performance-based selection. Most planning-based workflow composition systems tend to be limited as outlined in section 6.1.1. We wish to improve the flexibility by incorporating CBR and semantic-based technologies.

9.2 Automatic Output Rate Optimisation

Most video analyses are conducted manually using highly specialised tools for specific application areas. Using workflow technology for process automation, we aim to utilise the strengths of different image processing tools/operators. Thus we could optimise the output rate of automatic video processing algorithms and reuse them. Traditionally image processing experts spend many weeks or months developing these algorithms and do not reuse them.

10 Work in Progress

10.1 Ontology Population

Based on the framework provided, three ontologies have been provisionally created and populated. The basic concepts for the goal and domain ontologies were adopted from an extensive IP ontology provided by the Hermes project [7][1].

Goal Ontology. The goal ontology contains the high level goals and constraints that the user will communicate to the system. These are represented by the concepts `Goal, Constraint Category, Constraint Descriptor` and `Constraint Qualifier` in Fig. 5. The concept `Task/Process` links the goal to the processes that are associated with it. The instances of processes are contained within a process library [37] and will be selected based on task decomposition and performance criteria. These tasks will also be linked with the capability ontology.

Domain Ontology The domain ontology describes the concepts and relationships of the application area. This includes various aspects of the videos, such as the lighting conditions, colour information, position, orientation as well as spatial and temporal aspects. Fig. 6 illustrates these concepts in more detail.

Capability Ontology The capability ontology contains the classes of video and image processing techniques. Each technique (or capability) is associated with one or more tool. A tool is a software component that can independently perform a video or image processing task, or a technique within an integrated vision library that may be invoked with given parameters. Fig. 7 provides a graphical view of this ontology.

10.2 Walkthrough [39]

Based on the devised ontologies, a walkthrough on how they are used to provide different levels of vocabulary for the users, vision tools and processes in a seamless and related manner is outlined here. The user, who is an ecologist, may have a high level goal in mind, for instance *"Detect fishes in the video"*. This is represented and selected via the following selection-value pairs:

(Goal: Detection) [Detail Level: Occurrence = all_occurrences]
[Performance Criteria: Processing Time = real_time]

As a first step, the user interacts with the system by providing input values for the goal and constraints. The goal can be one of the goals specified in the goal ontology (Fig. 5). While the constraints are additional parameters to specify rules or restrictions that apply to the goal. These include qualifiers e.g. `Acceptable Errors, Optimisation Criteria, Detail Level` and `Quality Criteria` contained within the goal ontology.

Then the user describes the images to be processed. This description is given using the system which proposes descriptors contained in the domain ontology

---

[1] The ontology is available as `http://www.greyc.ensicaen.fr/~arenouf/Hermes/onto/ontoClient.owl`. Much appreciation goes to Arnaud Renouf for allowing us to use concepts from this ontology as a starting point.

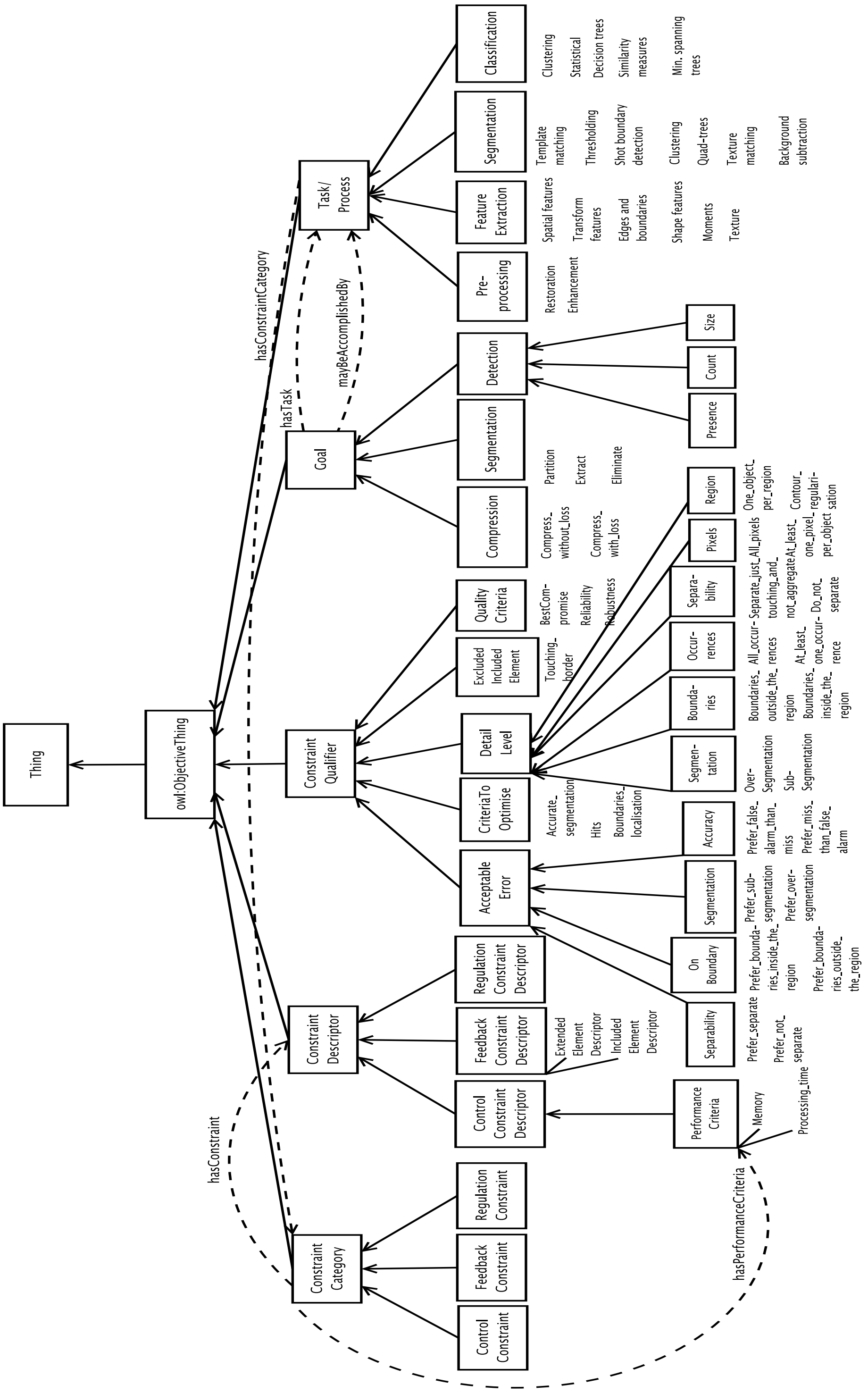


Figure 5: Goal Ontology.

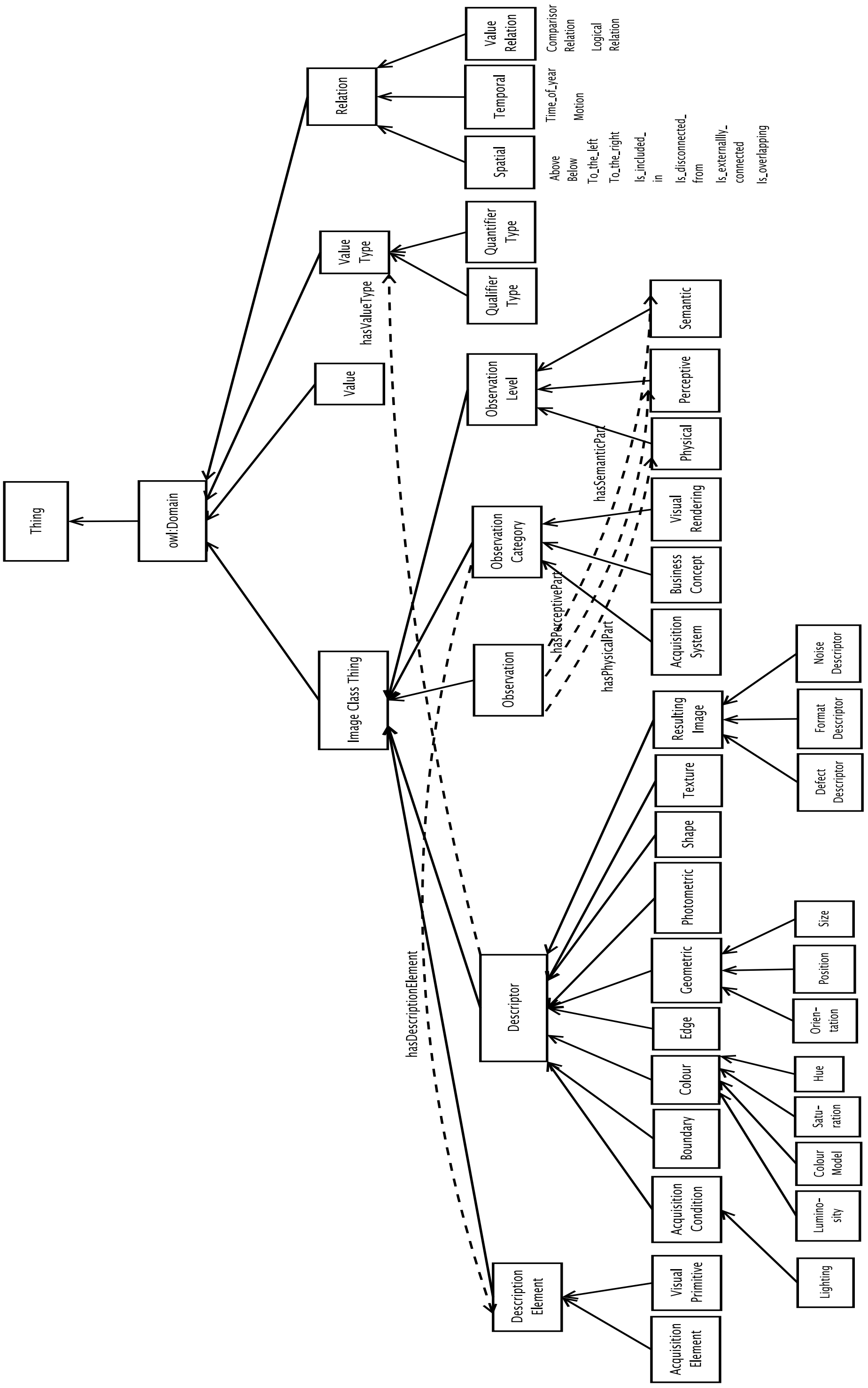


Figure 6: Domain Ontology.

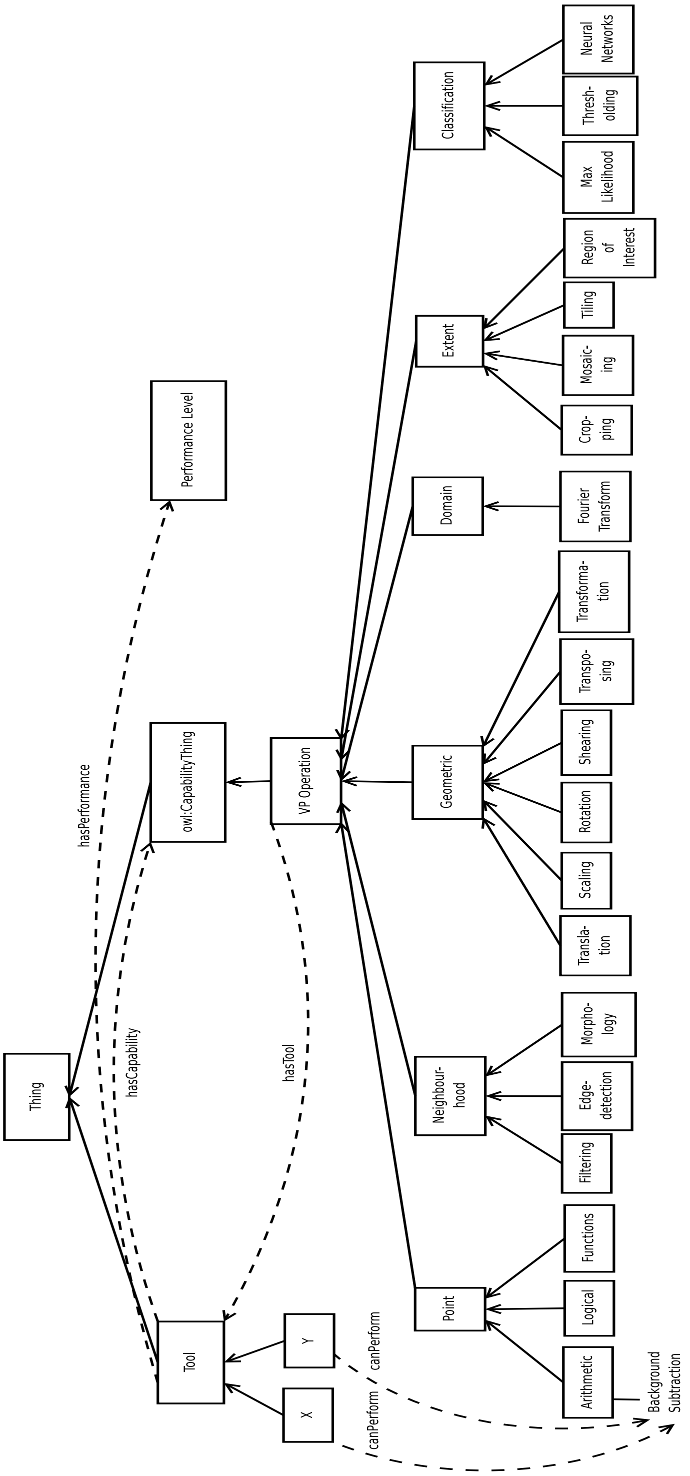


Figure 7: Capability Ontology.

(Fig. 6). In our scenario, a description on the acquisition context and on the semantic content of the images is obtained; the lighting conditions changes very slowly, the camera is fixed (and also the background), images are degraded by a blocking effect due to the compression, and the fishes are regions whose colours are different from the background and are bigger than a minimal area (because when they are too small they are unusable). Thus, this goal is interpreted as *"The detection of all the occurrences of non-background regions on a fixed background."*

As soon as the formulation of the user's problem is made, a sequence of processes for execution using task decomposition will be sought by a Planning mechanism. For instance, a Prolog-style syntax for the instances of processes in the library could be represented as follows:

```
detect_presence(X) :-
            pre_process(X), segmentation(X),
            classification(X).
pre_process(X) :-
            keyframe_detection, enhancement.
segmentation(fish) :-
            background_subtraction(fish).
background_subtraction(X) :-
            background_model_construction,
            model_differencing, background_model_update.
```

Each task within this sequence could be further decomposed into sub-tasks. For instance, the 'Segmentation' task involves 'Background Subtraction' (under `Task/Process` in the goal ontology), which is done in three steps; background model construction, current frame and background model differencing, and background model update. All these sub-tasks are point arithmetic operations in the capability ontology. When a sub-task can no longer be decomposed, the tool or technique identified for performing it can be applied directly on the video clip. The tool or technique available within the system is represented in the capability ontology. In the case where more than one tool is available to perform the sub-task, the tool with the best performing capability is selected.

The sequence of processes for execution constitutes the workflow composition. Should the Planner fail to find a solution for the workflow, the case library is consulted for the next closest solution possible. A manual implementation of this walkthrough is provided in [39].

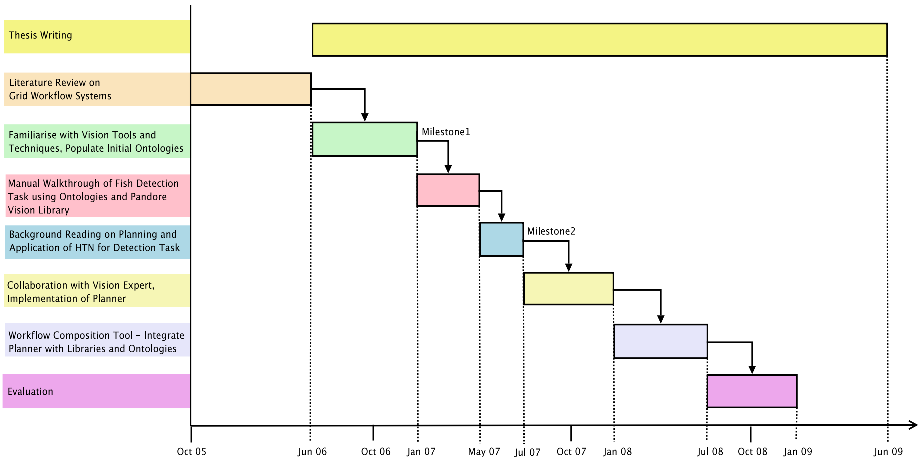
11 Risks

Throughout the duration of the research, several risks that could deter the work from being carried out according to schedule have been identified. Such risks could be technical, project-related or personal in nature. Appropriate measures should be taken to avoid them in advance, otherwise it is important to have contingency plans to work around these adverse conditions should they occur.

Technical Risks. As most software and libraries associated with the development of the work are open-source, there should not be major issues in obtaining the packages for download and installation. However, most open source libraries are prone to contain bugs which often cause compilation problems. These should be fixed with some knowledge in C/C++ which we have. The two libraries in question (Pandore and OpenCV) are extensive and contain IP functions that require some level of expertise in vision which we lack. Learning all the functionalities would be too time consuming and thus assistance is required from vision developers who are familiar with the techniques contained within the libraries. This assistance has been sought and received internally (IPAB) and externally (University of Catania and Laboratoire GREYC, Equipe Image).

12 Schedule/Work Plan

The duration of this research is 48 months from October 2005. The work plan for the project is as follows with several milestones:



- Milestone1: Goal, domain and capability ontologies. Vision tools and libraries acquired.
- Milestone2: A working example of Fish Detection using Pandore Vision Library. (submitted to CAIP 07)

13 List of Publications to Date

- Gayathri Nadarajan. "Planning for Video Processing using Ontology-Based Workflow" *Student Paper for Doctoral Consortium, ICAPS 07*, September 2007.
- Gayathri Nadarajan and Arnaud Renouf. "A Modular Approach for Automating Video Analysis". *CAIP 07*, August 2007.
- Gayathri Nadarajan and Yun-Heh Chen-Burger. "Translating a Typical Business Process Modelling Language to a Web Services Ontology through Lightweight Mapping". *IET Software*, February 2007.

- Gayathri Nadarajan, Yun-Heh Chen-Burger, James Malone. "Semantic Grid Services for Video Analysis." *SERCOMP'06*, December 2006.
- Gayathri Nadarajan, Yun-Heh Chen-Burger, James Malone. "Semantic-Based Workflow Composition for Video Processing in the Grid". *WI'06*, December 2006.
- Gayathri Nadarajan and Yun-Heh Chen-Burger. "Mapping Fundamental Business Process Modelling Language to OWL-S." *SETN'06*, May 2006.
- Gayathri Nadarajan and Yun-Heh Chen-Burger. "An Ontology-Based Conceptual Mapping Framework for Translating FBPML to the Web Services Ontology." *CCGrid'06*, May 2006.

References

[1] MATLAB - The Language of Technical Computing, The MathWorks Inc., 1994-2007. http://www.mathworks.com/products/matlab/.

[2] Project aceMedia: Integrating Knowledge, Semantics and Content for User-Centred Intelligent Media Services, 2004. http://www.acemedia.org/aceMedia.

[3] EcoGrid National Center for High Performance Computing, Taiwan, 2006. http://ecogrid.nchc.org.tw/.

[4] DAGMan Application, 2006. http://www.cs.wisc.edu/condor/manual/v6.8/2_11DAGMan_Applications.html.

[5] Open Computer Vision (OpenCV) Library 1.0, 2006. http://sourceforge.net/projects/opencvlibrary/.

[6] ORION: Intelligent environments for the resolution of problems for autonomous systems, 2007. http://www.inria.fr/recherche/equipes/orion.en.html.

[7] Hermès - a human-machine interface for the formulation of image processing applications, 2007. http://www.greyc.ensicaen.fr/~arenouf/Hermes/index-en.html.

[8] *BPM 2002: Market Milestone Report.* Delphi Group, Feb 2002.

[9] F. Berman, A. Chien, K. Cooper, J. Dongarra, I. Foster, D. Gannon, L. Johnsson, K. Kennedy, C. Kesselman, J. Mellor-Crumme, D. Reed, L. Torczon, and R. Wolski. The grads project: Software support for high-level grid application development. *Int. J. High Perform. Comput. Appl.*, 15(4):327–344, 2001.

[10] S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, V. Tzouvaras, Y. Avrithis, S. Handschuh, Y. Kompatsiaris, S. Staab, and M. G. Strintzis. Semantic Annotation of Images and Videos for Multimedia Analysis. In *ESWC*, volume 3532 of *LNCS*, pages 592–607. Springer, May 2005.

[11] V. Bombardier, P. Lhoste, and C. Mazaud. Modélisation et Intégration de Connaissances Métier pour L'identification de Défauts par règles Linguistiques Floues. *Traitement du Signal*, 21(3):227–247, 2004.

[12] BPEL4WS. *Business Process Execution Language for Web Services Version 1.1.* IBM, BEA Systems, Microsoft, SAP AG, Siebel Systems, 2003. http://www-128.ibm.com/developerworks/library/specification/ws-bpel/.

[13] J. Cardoso, J. Miller, A. Sheth, and J. Arnold. Modelling Quality of Service for Workflows and Web Service Processes. *Web Semantics Journal: Science, Services and Agents on the World Wide Web*, Dec 2004. Vol 1, No. 3, pp. 281–308.

[14] G. A. Cesare Pautasso, Thomas Heinis. JOpera: Autonomic Service Orchestration. *IEEE Data Engineering Bulletin (Special Issue on Self-Managing Database Systems)*, Sept 2006. Vol. 29 No. 3.

[15] Y.-H. Chen-Burger and F.-P. Lin. A Semantic-based Workflow Choreography for Integrated Sensing and Processing. In *Proceedings of the 9th IEEE International Workshop on Cellular Neural Networks and their Applications*, 2005.

[16] S. Chien and H. Mortensen. Automating Image Processing for Scientific Data Analysis of a large Image Database. *IEEE PAMI*, 18(8):854–859, august 1996.

[17] V. Clément and M. Thonnat. A Knowledge-Based Approach to Integration of Image Procedures Processing. *CVGIP: Image Understanding*, 57(2):166–184, Mar 1993.

[18] R. Clouard, A. Elmoataz, and F. Angot. Une Bibliothèque et un Environnement de Programmation d'opérateurs de Traitement d'images. In *Rapport interne du GREYC, Caen, France*, Mar 1997.

[19] R. Clouard, A. Elmoataz, C. Porquet, and M. Revenu. Borg : A Knowledge-Based System for Automatic Generation of Image Processing Programs. *IEEE PAMI*, 21(2):128–144, 1999.

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

[21] B. Draper, A. Hanson, and E. Riseman. Knowledge-directed vision : Control, learning, and integration. In *Proc. of IEEE*, volume 84, pages 1625–1681, 1996.

[22] T. Fahringer, R. Prodan, R.Duan, J. Hofer, F. Nadeem, F. Nerieri, S. Podlipnig, J. Qin, M. Siddiqui, H.-L. Truong, A. Villazon, and M. Wieczorek. ASKALON: A Development and Grid Computing Environment for Scientific Workflows. *Workflows for eScience, Scientific Workflows for Grids*, 2007.

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

[24] I. Foster, J. Voeckler, M. Wilde, and Y. Zhao. Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation. In *14th Conference on Scientific and Statistical Database Management*, 2002.

[25] K. Gupta and A. Montezemi. Empirical Evaluation of Retrieval in Case-Based Reasoning Systems using Modified Cosine Matching Function. In *IEEE Trans. Systems, Man and Cybernetics*, 1997. Vol. 27(5), pp. 601–612.

[26] C. Gutierrez, C. Hurtado, and A. Vaisman. Temporal RDF. In *Proceedings of European Conference on the Semantic Web (ECSW'05)*, Jan 2005.

[27] C. Hudelot. *Towards a Cognitive Vision Platform for Semantic Image Interpretation. Application to the Recognition of of Biological Organisms.* PhD thesis, Nice-Sophia Antipolis University, 2005.

[28] M. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based Multimedia Information Retrieval: State-of-the-art and Challenges. *ACM Transactions on Multimedia Computing, Communication, and Applications*, Feb 2006. Vol 2, No. 1, pp. 1–19.

[29] C. Liedtke and A. Blömer. Architecture of the Knowledge Based Configuration System for Image Analysis "Conny". In *ICPR'92*, pages 375–378, 1992.

[30] X. Liu, J. Liu, J. Eker, and E. Lee. Heterogeneous Modeling and Design of Control Systems. *Software-Enabled Control: Information Technology for Dynamical Systems*, April 2003.

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

[32] N. Maillot, M. Thonnat, and A. Boucher. Towards Ontology Based Cognitive Vision. *Machine Vision and Applications (MVA)*, Dec 2004. Vol 16, No. 1, pp. 33–40.

[33] T. Matsuyama. Expert Systems for Image Processing: Knowledge-based Composition of Image Analysis Processes. *Computer Vision, Graphics, and Image Processing*, Oct 1989. Vol. 48 No. 1. pp. 22–49.

[34] S. McGough, L. Young, A. Afazal, S. Newhouse, and J. Darlington. Workflow Enactment in ICENI. In *UK e-Science All Hands Meeting, Nottingham, UK*, Sept 2004. pp. 894–900.

[35] D. McGuinness and F. van Harmelen. *OWL Web Ontology Language*. World Wide Web Consortium (W3C), 2004. http://www.w3.org/TR/owl-features/.

[36] G. Nadarajan. Planning for Video Processing using Ontology-Based Workflow. In *Student Paper for Doctoral Consortium at the 17th International Conference on Automated Planning & Scheduling (ICAPS 07), Providence, Rhode Island, USA*, Sept 2007.

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

[38] G. Nadarajan, Y.-H. Chen-Burger, and J. Malone. Semantic Grid Services for Video Analysis. In *Web Intelligence and International Agent Technology Workshops, Hong Kong*, Dec 2006. pp. 121–124.

[39] G. Nadarajan and A. Renouf. A Modular Approach for Automating Video Processing. In *12th International Conference on Computer Analysis of Images and Patterns (CAIP'07), Vienna, Austria*, Aug 2007.

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

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

[42] J. Rao and X. Su. A Survery of Automated Web Service Composition Methods. In *Proceedings of First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), San Diego, CA, USA*, 2004.

[43] A. Renouf, R. Clouard, and M. Revenu. A Platform Dedicated to Knowledge Engineering for the Development of Image Processing Applications. In *9th International Conference on Enterprise Information Systems (CEIP 2007)*, June 2007.

[44] A. Renouf, R. Clouard, and M. Revenu. How to Formulate Image Processing Applications? In *International Conference on Computer Vision Systems (ICVS 2007)*, Mar 2007.

[45] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel. Web Services Modelling Ontology. *Applied Ontology*, 2005. Vol. 1 No. 1. pp. 77–106.

[46] D. D. Roure, N. R. Jennings, and N. R. Shadbolt. The Semantic Grid: Past, Present, and Future. In *Proceedings of the IEEE Volume 93, Issue 3, pp. 669–681*, March 2005.

[47] R. Stevens, A. Robinson, and C. Goble. MyGrid: Personalized Bioinformatics on the Information Grid. *Bioinformatics*, 2003. Vol. 19, Suppl. 1, pp. i302–i304.

[48] T. Tannenbaum, D. Wright, K. Miller, and M. Livny. Condor – a distributed job scheduler. In T. Sterling, editor, *Beowulf Cluster Computing with Linux*. MIT Press, Oct 2001.

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

[50] I. Taylor, M. Shields, I. Wang, and A. Harrison. Visual Grid Workflow in Triana. *Journal of Grid Computing*, September 2005. 3(3-4):153-169.

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