

A Knowledge-Based Planner for Processing Unconstrained Underwater Videos



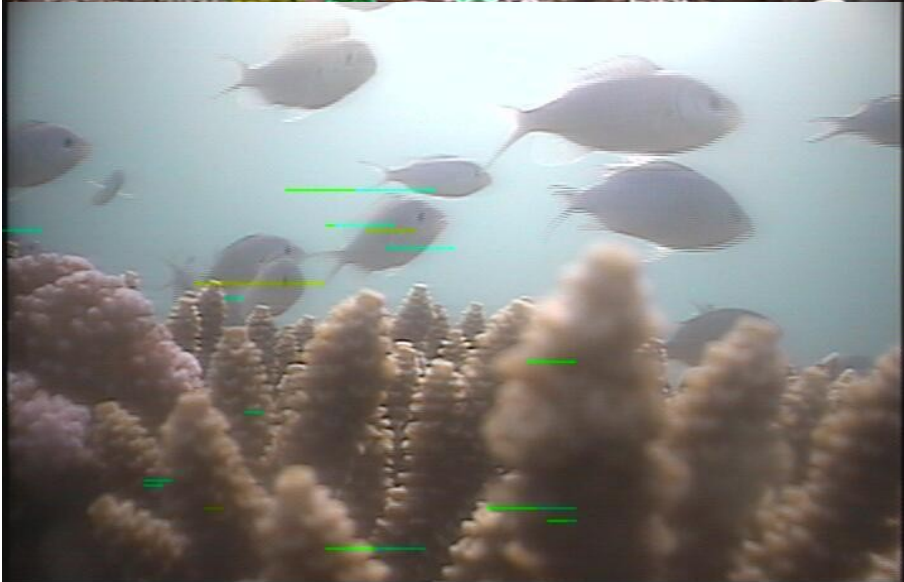
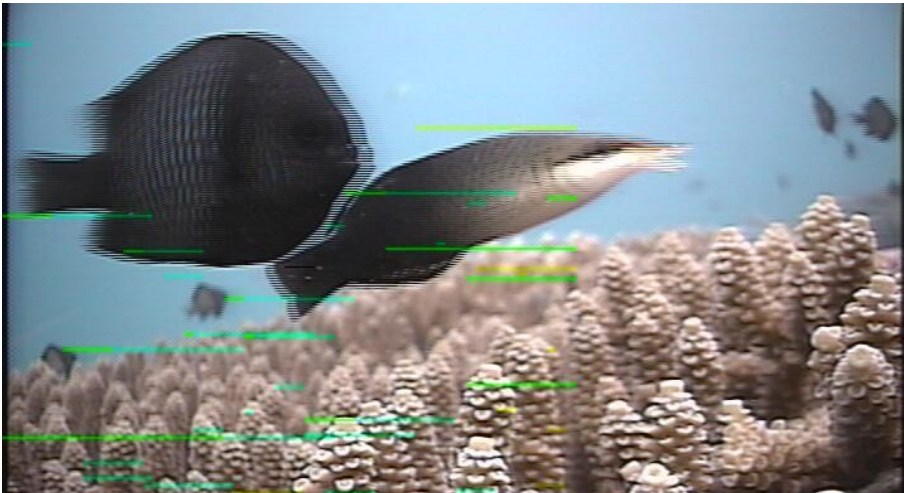
Gaya Nadarajan, Jessica Chen-Burger, Bob Fisher

University of Edinburgh, U.K.

Introduction

- Video data collected continuously is pervasive today
- Analyzing them in an efficient manner is a challenge
- Ecogrid scenario – underwater videos of varying quality are captured and stored for analysis.

Sample captures from videos



video2

2007-08-02 07:52:00

July 12th 2009



video1

2007-04-29 13:10:00

HTN Planning for Processing Underwater Videos

Problem

- A one minute video clip typically takes 1500+ frames and is stored in 3–4 MB. This translates into 200+ MB per minute, 5+ GB per day and 2 TB per year for one operational camera
- Manual analysis is too time consuming:

One minute's clip will cost approx. 15 minutes' manual processing time → One year's recording of a camera would cost 15 years' effort just to perform basic analyzing and classifying tasks

Impractical situation and appropriate **automated** means should be introduced

Requirements

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

Existing Efforts

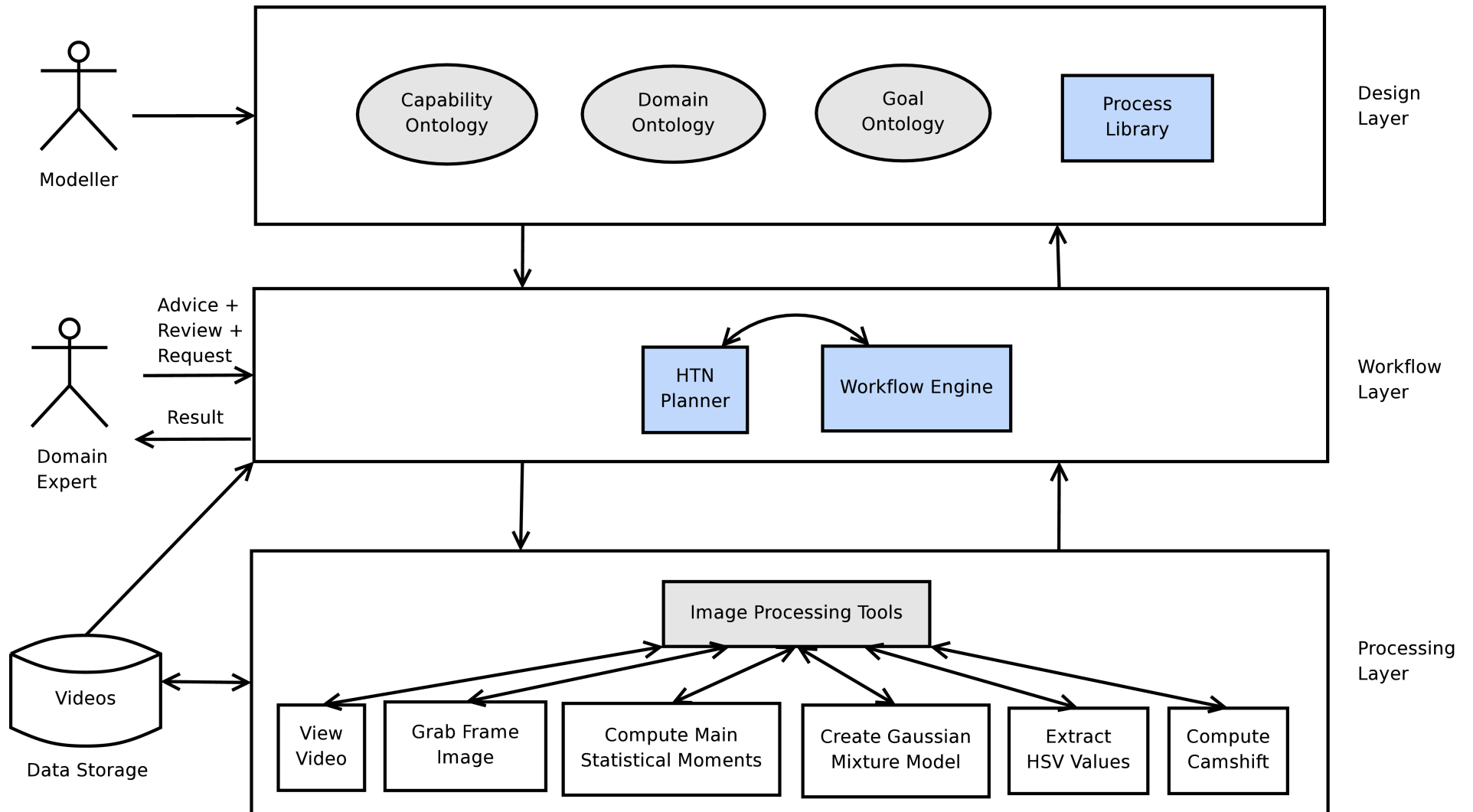
- Automatic image processing efforts started from the 80's with expert systems and spanned to the 90's (OCAPI, MVP, BORG) and more recently knowledge based systems (Orion, aceMedia)

Pitfalls: - Restricted to well-known goals
- *A priori* knowledge about the domain (e.g. noise, lighting) were not taken into account

- Workflow systems also emerged as suitable solutions for providing automation and flexible mechanisms for processing repetitive, tedious tasks (Pegasus, Taverna, Kepler, Triana)

Pitfalls: - Require user to possess technical knowledge in order to compose workflow (e.g. Bioinformaticians in Taverna)

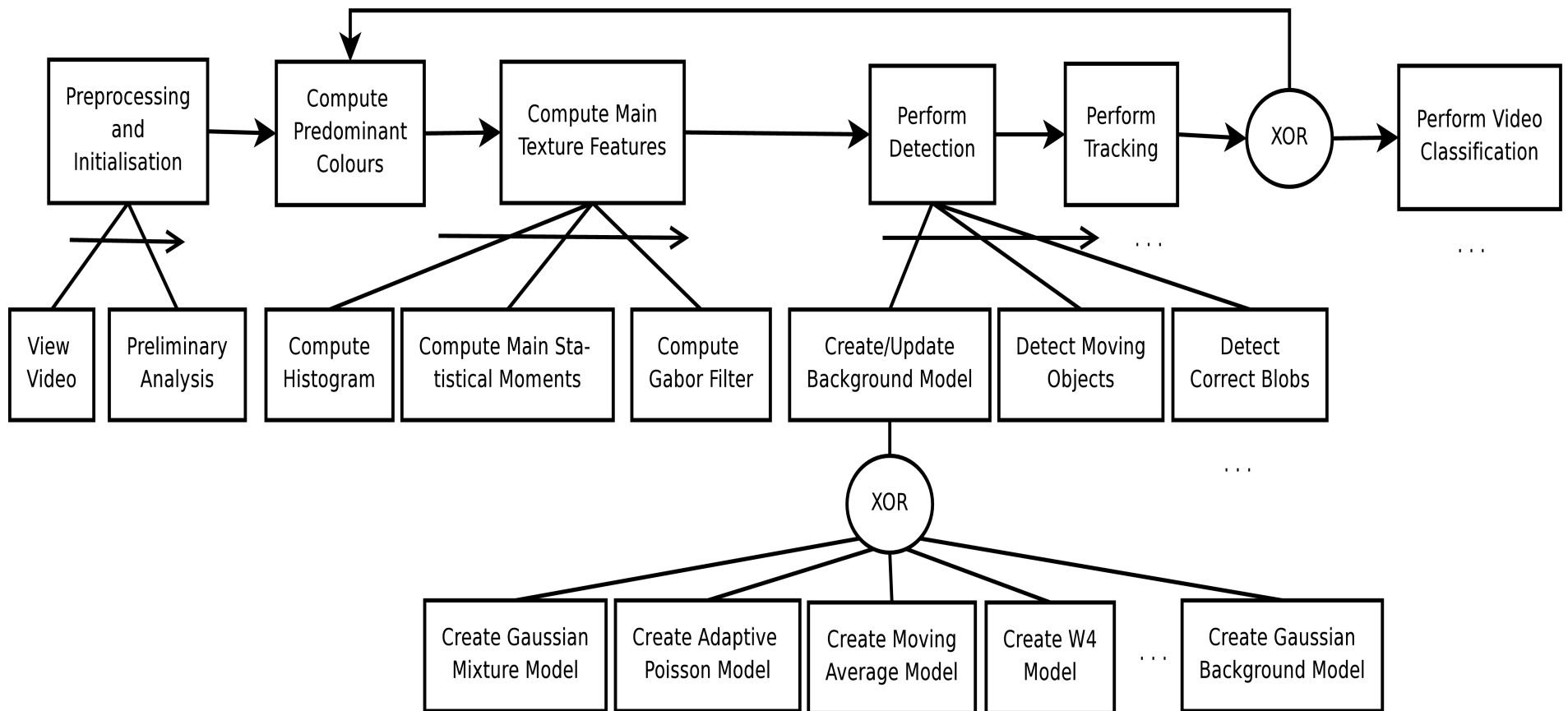
Proposed Framework



HTN Planning

- A video processing task is seen as a planning problem where the goal is to perform a set of tasks
- Tasks are **decomposed into sub-tasks exhaustively until primitive tasks are encountered**
- HTN planning provides a fast and efficient way of planning without having to explore the whole search space blindly as **heuristics** are encoded as methods that will prune the search tree considerably as only the applicable methods for the current goal (subtask) will be considered
- Adopts a total order SHOP-style algorithm
- Methods encoded in process library

Classify Video, Detect Fish, Count Fish



Methodology

- Video processing program was divided into **independent executables** or tools, each able to perform some primitive operation. Examples:
Grab Frame Image, Compute Gabor Filter, Extract HSV Values, Compute Histogram Backprojection, Compute Camshift
- The planner, given the goal (e.g. classify video) and initial domain description, sets out to find the steps taken to achieve this goal
- The resulting plan is presented to the user who may change domain and constraints, in which case replanning will take place.

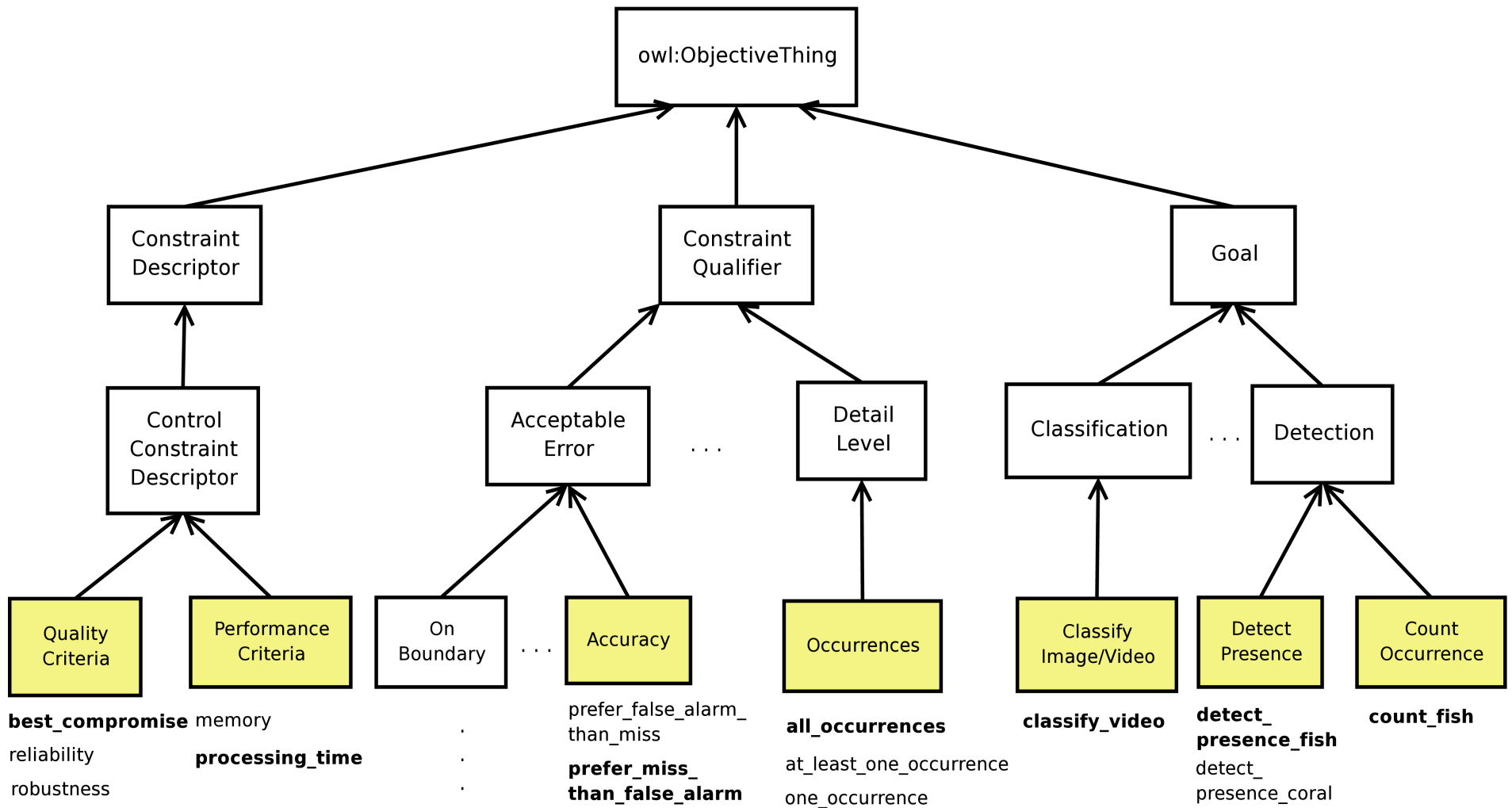
Ontologies

- Formal conceptualisations of entities that are relevant to a particular problem domain so that these representations and the relationships between them are made explicit
- **Goal** ontology: high-level video processing tasks (goals) and constraints that the user will communicate to the system
- **Domain** ontology: concepts and relationships associated with the videos, such as the lighting conditions, colour information, position, orientation as well as spatial and temporal aspects
- **Capability** ontology: classes of video and image processing tools, their functionalities and *performance indicators* tied to the domain description and/or constraints.

Walkthrough

- User provides high level goal (default - classify video)
- User provides initial domain description and constraints (generated automatically otherwise based on suggested values)
- Goals (contained in goal ontology)
 - Classify Video
 - Detect Fish
 - Classify Video, Detect Fish, Count Fish
- Constraints (contained in goal ontology)
 - Performance (memory vs processing time)
 - Quality (reliability, robustness, best_compromise)
 - Accuracy (prefer_false_alarm than miss, prefer miss than false alarm)
 - Occurrence (all, at_least_one)
- Domain Description (contained in domain ontology)
 - Brightness (dark, medium, bright)
 - Clearness (clear, blur)
 - Percentage Background Object (low, medium, high)

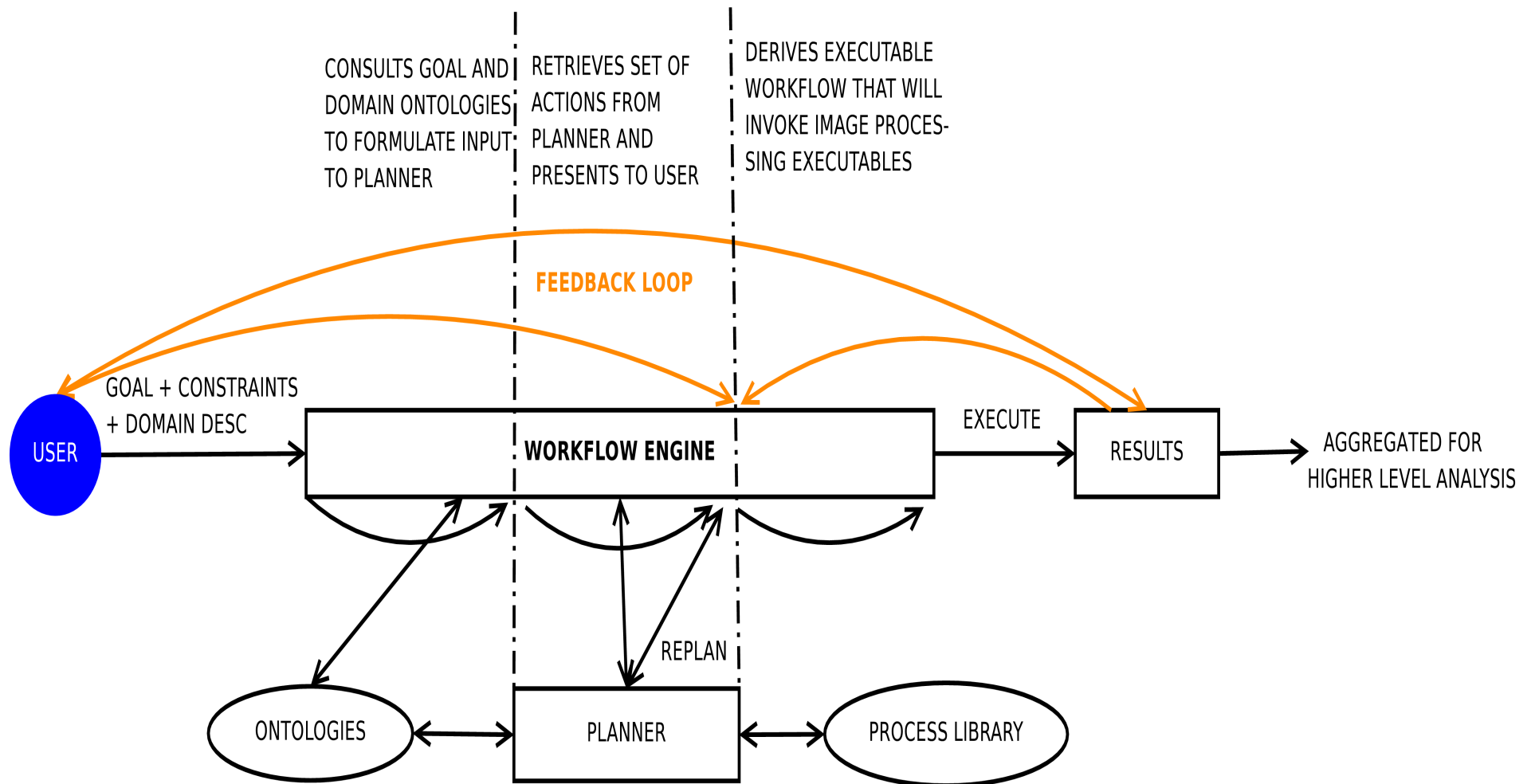
Goal Ontology with Relevant Concepts



Walkthrough cont.

- Workflow, with the aid of goal and domain ontologies, formulates goal as input to planner
- Planner, with the aid of process library which contains methods and capability ontology selects best combination of tools to solve task
- User is prompted for confirmation before plan is sent for execution, user may modify domain information and constraints, in which case replanning will take place
- After execution, result is presented to user for evaluation, hence performance of system can be improved incrementally

User-System Interaction



Deriving HTN Plans from User Knowledge and IP Heuristics

- **Background Model Creation**

7 types of background models to choose from

- **Gaussian Background Model**

Clearness = high, Speed of movement = low, % Background Object = low, Noise Type = Gaussian

- **Gaussian Mixture Model**

Speed of movement = high, % Background Object = medium, Noise Type = Gaussian, Accuracy = prefer miss than false alarm

- **Moving Average Model**

Accuracy = prefer false alarm than miss, Performance = processing time

- **W4 Model**

% Background Object = high, Noise Type = Salt and Pepper

Sample Captures for Classify Video, Detect Fish, Count Fish



In each frame the text on the right hand side indicate (top-bottom) *Brightness*, *Clearness*, *Presence of Fish* and *Presence of Algae*. The numbers on the left indicate (top-bottom) *Number of Fish in Current Frame* and *Number of Fish in Video So Far*. The bounding boxes indicate objects that have been identified as fish

Implementation and Results

- Workflow, planner and process library implemented in Prolog to allow for knowledge to be organised qualitatively and defined declaratively
- Ontologies are represented using a visual FOL-based language (FBPML-DL) which is importable to OWL
- 31 independent executables have been identified and developed using C++ based OpenCV library (an improvement from 85 previously)
- A sample run for classify video, detect fish and count fish task has yielded 1456 steps in 0.44 seconds for a video of 50-frames

Issues and Conclusions

- Integrating HTN planning with ontologies within a workflow system is indeed useful to provide assistance to non-specialists conduct video processing tasks in a time-efficient manner
- Performance of system will need to be improved incrementally, as result will not match specialized tailor-made IP solutions
- Future work - fish classification

Thank you!



video1

2007-04-29 17:20:00

gaya.n@ed.ac.uk