Developing Smart Spatial Systems Using CommonKADS

Paul Crowther

School of Computing and Management Sciences
Sheffield Hallam University
Telephone: +44 (0)114 225 2900
Email: P.Crowther@shu.ac.uk

Jacky Hartnett

School of Computing
University of Tasmania
Email: J.Hartnett@utas.edu.au

Abstract

The aim of this paper is to show how spatial reasoning systems in a geographic context can be developed using CommonKADS. KADS was initiated as a "Structured methodology for the development of knowledge based systems" (Motta. 1997). The limitations of production rules, combined with their inherent non-reusability contributed significantly to the impetus to develop methodologies like KADS. The two central principles that underlie the KADS approach are the introduction of multiple models as a means of coping with the complexity of the knowledge engineering process, and the use of knowledge-level descriptions as an intermediate model between expertise data and system design.

Motta (1997) coins the term “knowledge modelling revolution”, which refers to the paradigm switch from symbol level (rule based) approaches to knowledge level task centred analysis. This heralded the necessary decoupling of the task specification and the problem solving method.

The CommonKADS methodology has been used extensively in a wide range of domains, but not that of spatial reasoning. The advantage of CommonKADS is that it provides a full model of an application, not just a model of the knowledgebase (Schriber et al 2000) .

Visual geographic knowledge which can be extracted from satellite remote sensing images has characteristics which are not commonly found in non-visual domains. Traditionally geographic expert systems have worked either at the pixel level of raster images or the object level of vector images. This has shortfalls when knowledge acquired from a human image interpreter has to be incorporated into an expert system to aid interpretation.

1. Introduction

CommonKADS has been used in a wide variety of domains, but, rarely in the spatial domain. CommonKADS provides a series of models which allow the mapping of the complex domains found in knowledge based systems. Specifically one starts with the Organisational model to define the objectives of the system and organisational context in which it will be used, then develop other models to meet these objectives.

In this paper a series of CommonKADS models have been developed to describe a domain involving crop identification from satellite imagery. This project was of interest to local crop processing companies who had a number of requirements. Initially, simply producing classified images for these companies (Barrett et al 2000) was seen as ‘useful’. However no real requirement modelling apart from what will be described as the knowledge model was done. This became apparent after the results were shown to the companies who then had a series of different ideas of what they wanted from the system. For example one company was specifically interested in yield prediction. Another company wanted to identify all paddocks that could potentially be used for growing their crop.

The system that was originally developed was in reality a demonstrator. Each company had then to be looked at individually and their requirements, users and in some cases specific knowledge, modelled. CommonKADS was used as the basis of this modelling. This paper will present the methods used to develop specific applications with examples of the various CommonKADS models

2. CommonKADS

Figure 1 gives an overview of the interconnecting CommonKADS models. Each of the models has a series of associated worksheets which when filled in provide detailed documentation of the system and its requirements (Schreiber et al, 2000**)**. In this paper only the organizational, task, agent and knowledge model will be discussed in detail as the communication and design model become very application and platform specific.

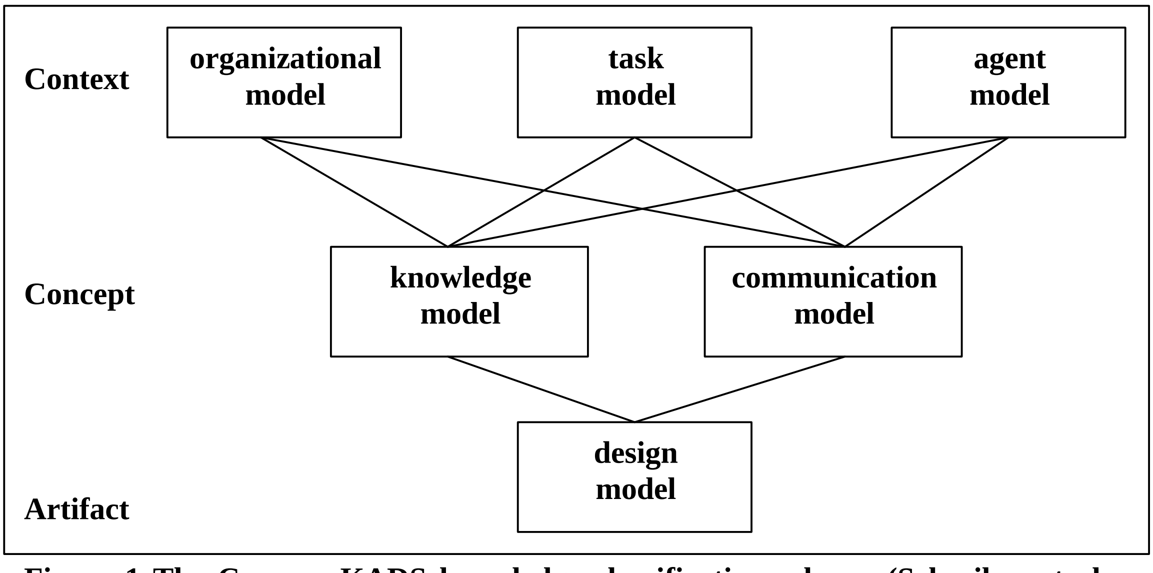


Figure 1 The CommonKADS knowledge classification scheme (Schreiber et al, 2000, p18)

The *Organisational Model* is a model which documents the objectives of the system and identifies opportunities of value to the organisation. One of the advantages of CommonKADS is that the organizational model provides an analysis of the socio-organizational environment that the KBS (Knowledge Based System) will have to function in. This includes descriptions of functions within the organization (Wielinga and Schreiber, 1993). The organisation model is also used to identify risks of fielding a KBS. These risks tend not to be technical as KBS's are now widely used and there is less risk of KBS failure through technical issues (De Hoog et al, 1993). This is as true of spatial KBS's as non-spatial ones.

The *Agent Model* provides an understanding of the systems users and identifies how these users or agents will perform their tasks (Gustafsson and Menezes 1994). An agent in this context is a person, hardware or software which interfaces with the KBS. In a spatial system, for example, it would be likely that an agent could be a GIS or an image sensor.

The CommonKADS *Task Model* specifies how the functionality of the system is achieved (Gustafsson and Menezes 1994). The task model links to the agent model to identify the people (or the roles that individuals have), hardware or systems that perform a task, to operate in the domain defined in the organizational model.

The *Communication Model* models the interaction of the system with the user and other system components. It models how software and users of the system work together and specifies the environment in which KBS must work.

The *Knowledge Model* (Figure 2) defines the knowledge necessary to meet the objectives specified in the organisational model and the tasks in the task model. It is split into three layers.

The CommonKADS *Domain Layer* is knowledge describing a declarative theory of the domain. Knowledge at this level should be represented in a way that is independent of the way in which it is to be used. Generally this is described using UML (Unified Modelling Language) notation (Rumbaugh et al 1999). It defines the conceptualisation and declarative theory of the problem domain and provides the knowledge to carry out given tasks. In other words it contains facts, rules and domain types. The other layers contain knowledge to control the use of knowledge from the domain layer (Fensel and Van Harmelen, 1994).

The *Inference Layer* specifies how to use the knowledge from the domain layer. It restricts the use of the domain layer and abstracts from it. A formal specification language has been developed to record knowledge in each of the layers (Schreiber et al 1994).

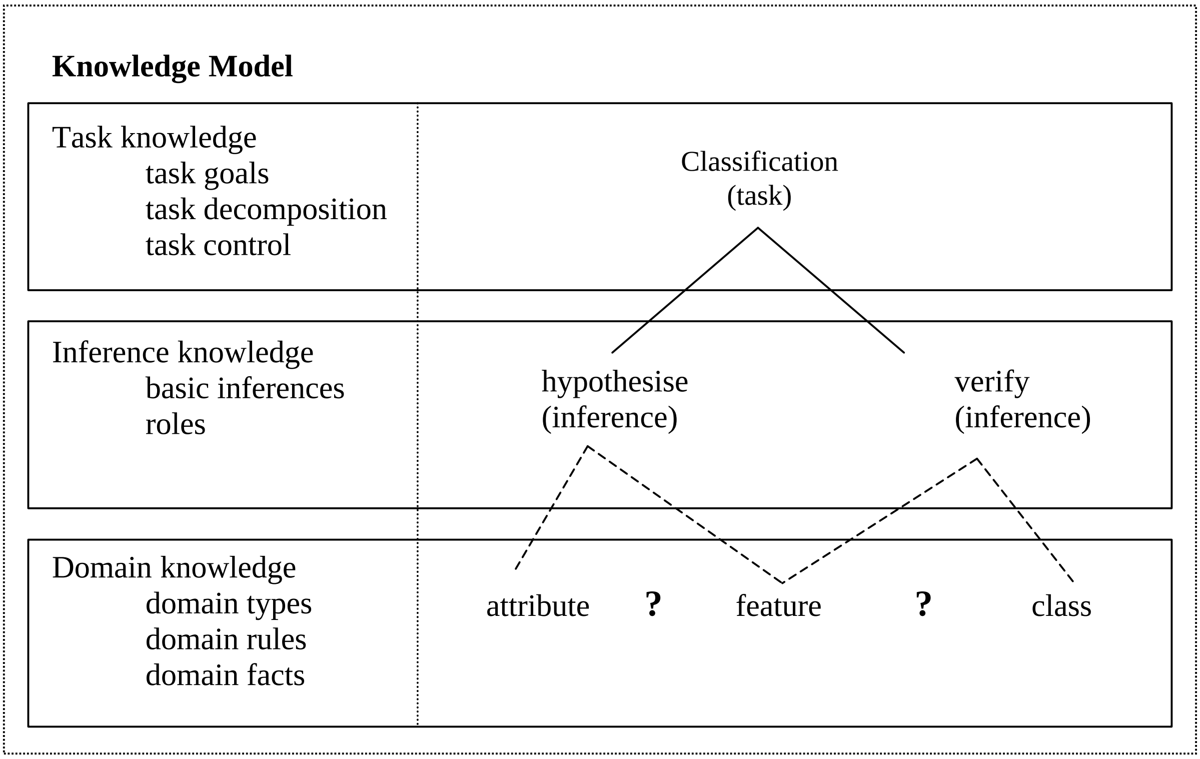


Figure 2: The three-layer knowledge model (after Schreiber et al, 2000).

The *Task Layer* represents a fixed strategy for achieving problem solving goals. It includes a specification of the goals related to a specific task and indicates how these goals can be decomposed into subgoals.

The task layer in the geographical classification scheme shows how to apply the problem solving strategy to a whole image set. Depending on the objective of the system, different strategies can be used, including the masking out of areas not relevant in a particular domain.

For example, in a geographical knowledge classification scheme, knowledge about specific objects, like an instance of the *crop* class *potato*, would be held at the domain level. Inference knowledge would be used to apply the domain knowledge At the task level there would be a task to identify all *paddocks* containing *potato crops.* This could be defined in more abstract terms (apply crop labels to paddocks) and reused to identify other crops.

The original KADS methodology proposes a four layer knowledge model (Schreiber et al, 1993). The CommonKADS methodology (Schreiber et al, 2000) amended this to the three layer model which is shown in Figure 2.

In the earlier 4 layer model, the top level of knowledge was the *Strategy Layer* which involves knowledge of how to choose between various tasks that when completed successfully achieve the same goal. This layer has been removed in the CommonKADS model, but may have application in spatial systems. In geographic knowledge applications the strategy layer could contain alternate ways of classifying images. This could be important when choices need to be made between using rules developed from machine learning, neural nets or rules elicited from domain experts to classify images. It should be noted once again that the strategy layer is not part of the CommonKADS model

3. A Spatial Application

The rest of this paper will present an example based on the interpretation of satellite imagery in an agricultural domain where expert knowledge is captured and applied. A set of CommonKADS models will be presented and evaluated.

In 1997 a project was initiated to create a crop identification and monitoring system using remotely sensed images from both the Landsat and SPOT satellite systems over the major agricultural region of North West Tasmania (figure 3). The aim was to establish crop distributions, monitor crop health and gain an estimate of yield. Crops of interest were restricted to poppies, potatoes, peas, pyrethrum and onions, all of which are grown for processing and export.

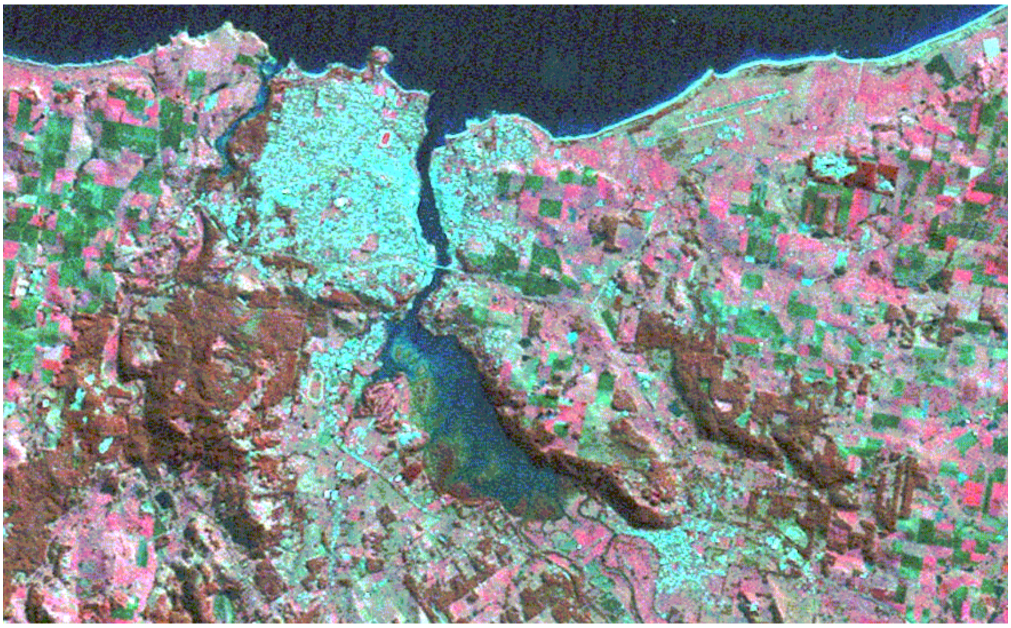


Figure 3 composite image of part of the study area around Devonport in North West Tasmania

The Landsat provided seven spectral bands of information and the SPOT image, three. All images were registered and resampled to cover the same area at 20 meters resolution. Various approaches to crop classification were carried out using clustering algorithms, neural networks and an expert systems approach. The chosen approach giving most consistent results was the method using clustering algorithms (Barrett et al, 2000)

Figure 4 shows a UML class model (Rumbaugh et al 1999) defining the overall problem domain. This model forms the design for the database, which is part of the system. The automated part of the system deals with crop identification. Some research has been carried out into automated boundary (fence) extraction (Anderson et al, 2000), but at the moment this is not reliable enough to use. Paddock boundaries have therefore been manually digitized and added to the system as an overlay. A soil distribution map and a digital elevation model can also be added as a map overlay and linked to paddock once the boundary digitization has been completed. Classes such as *farmer*, *agRegion*, *fertilizer* and *spray* are all entered manually. Once entered, these can be displayed as overlays based on their association with the class *paddock*.

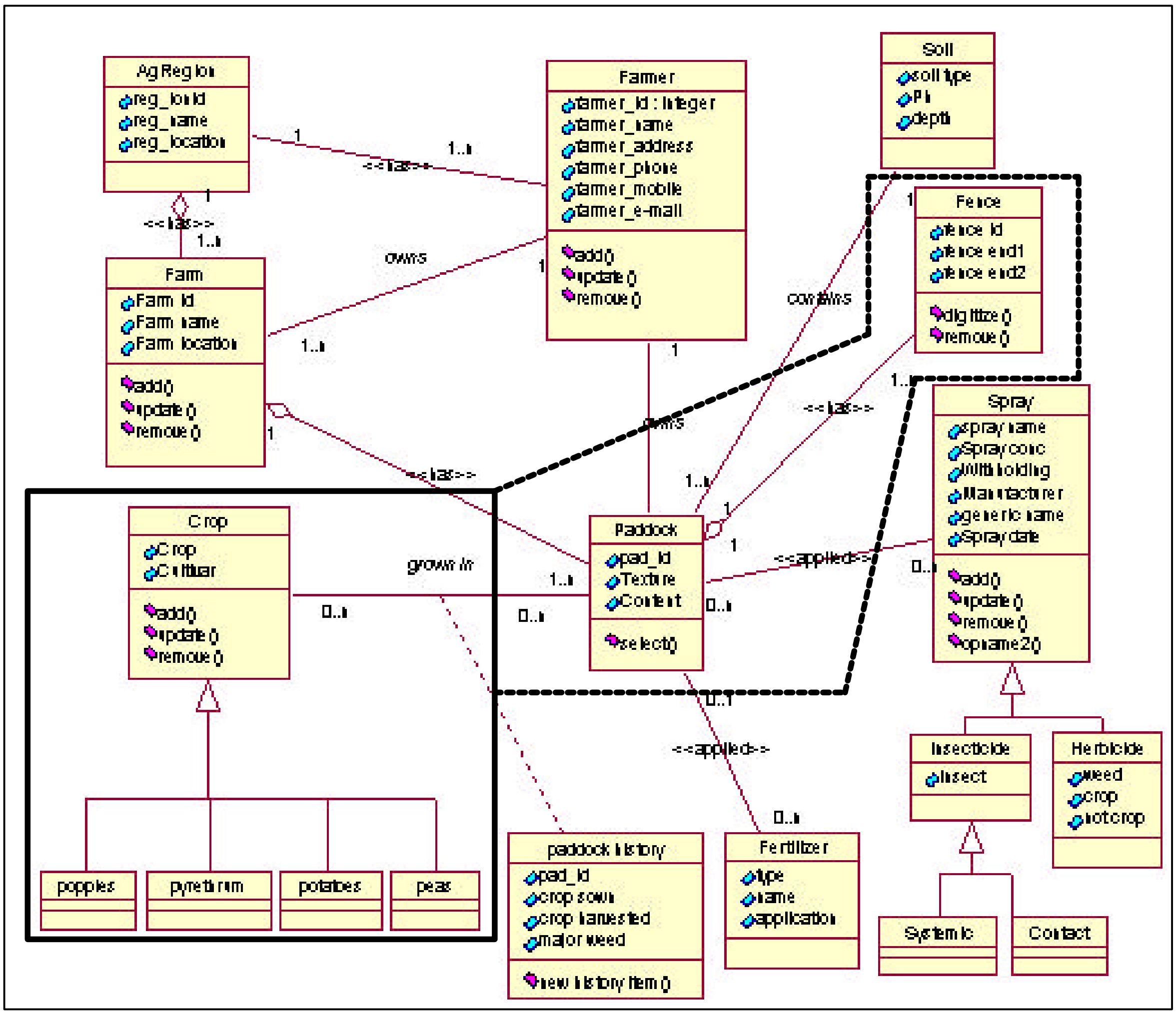


Figure 4. Extended domain required for use in agriculture. The polygonal items (solid line) are within the current problem domain. Fence (or more accurately paddock edge) and paddock (within the dotted line) could be part of the visual domain but currently are digitised, rather than identified by reasoning.

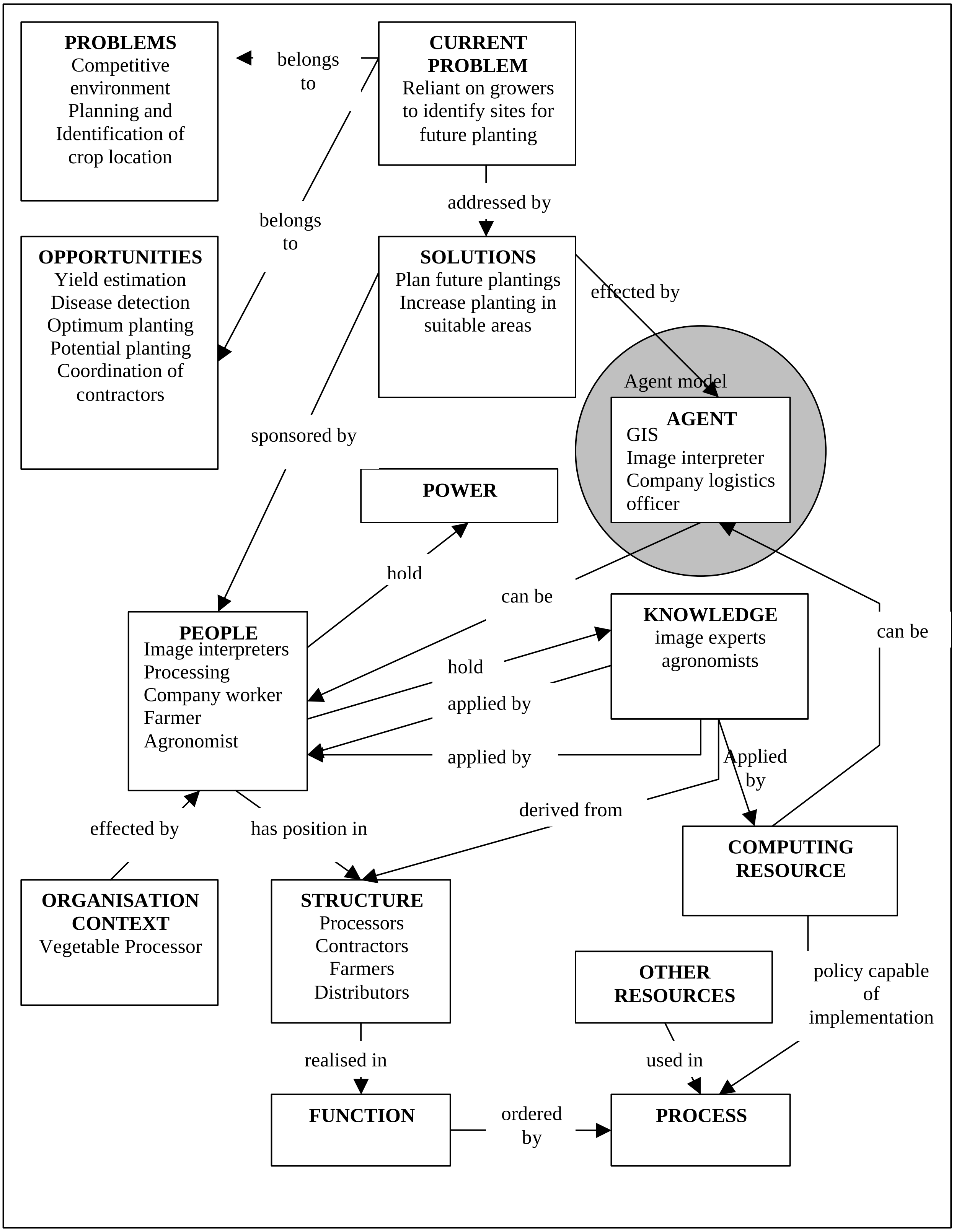


Figure 5 Organisational model

A number of objects that are of interest to the system have been extracted as part of the model. Two of the objects have spatial characteristics. *Paddock* and *Crop* are both polygonal objects. A *paddock* contains a *crop,* although at any one time there is only one *crop* in a *paddock*. The different types of crop of interest to the case study have been shown as specialisations of *crop* and inherit all of *crop*'s attributes and operations. Specific attributes can be added to these specialisations as required. In this system it is the *crop* which is identified and then assigned to the *paddock* which effectively is added as an overlay.

In the initial stages of the study, the crop identification techniques were paramount, with only a broad definition of how the final system would be used. This study addresses that problem by building models for individual applications using CommonKADS models. It should however be noted that many of the components of these models can be reused (Valente, 1998).

3.1. Organisational model.

Figure 5 is a skeleton of the organisation model for a vegetable processing company that wants to use a satellite image classified by crop. This establishes the organizational setting, problems, opportunities and available resources. For example one requirement within the company is to know the status of the current seasons crop. A second requirementa is where that crop could be grown the following season which is governed by what was grown in the previous season. This lead to the requirement of recoding the *paddock history* (figure 4*)*.

The aim of producing an organisation model is to establish the objectives of the system in terms of the organisation (De Hoog, 1994). In this case a specific requirement of determining where to plant was established.

3.2. Task Model

The agent and the task model are shown in together figure 6 in the form of a UML use-case diagram. The use cases describe the tasks which are required. The primary task of the system as it was initially developed was to produce a classified image. This is still required, but a classified image on its own is of limited value to the organisation.

Therefore, from the Organisational Model view, the primary task is to identify the location where crops can be sown in the following season. This task also requires knowledge of crop rotation which will be found in the domain layer of the Knowledge Model as well as knowledge about how to identify current crops.

There are also a series of other tasks which are required before the primary task can be completed. For example, to be able to identify current crops, an image set needs to be loaded, then classified.

Finally there needs to be tasks which provide a mechanism to add knowledge to the system which can be used to produce that classification. Each task in the Task Model is described by a template (Schreiber et al, 2000), an example of which is shown below:

TASK: 02 Identify next seasons planting location identification
ORGANIZATION agronomist, contractor coordinator, strategic crop planner
GOAL AND VALUE identify paddocks which where the crop can be grown
Find optimal locations which will result in a good crop yield
DEPENDENCY AND FLOW
Input tasks: specific crop type distribution,
Output tasks: paddock location for sowing
OBJECTS HANDLED
Input objects: classified image (showing crop which precedes the target planting crop)
Output objects: classified image with target paddocks for sowing
Internal objects: this is shown by figure 4
TIMING AND CONTROL
Frequency: yearly.
Timing: after collection of growing season images and before planting time
(i) *preconditions* crop ready for harvesting;
(ii) *postconditions* paddocks ready for sowing.
AGENTS agronomist, GIS
KNOWLEDGE AND COMPETENCE Competent with GIS
RESOURCES As required
QUALITY AND PERFORMANCE Improve the total of harvest per area harvested

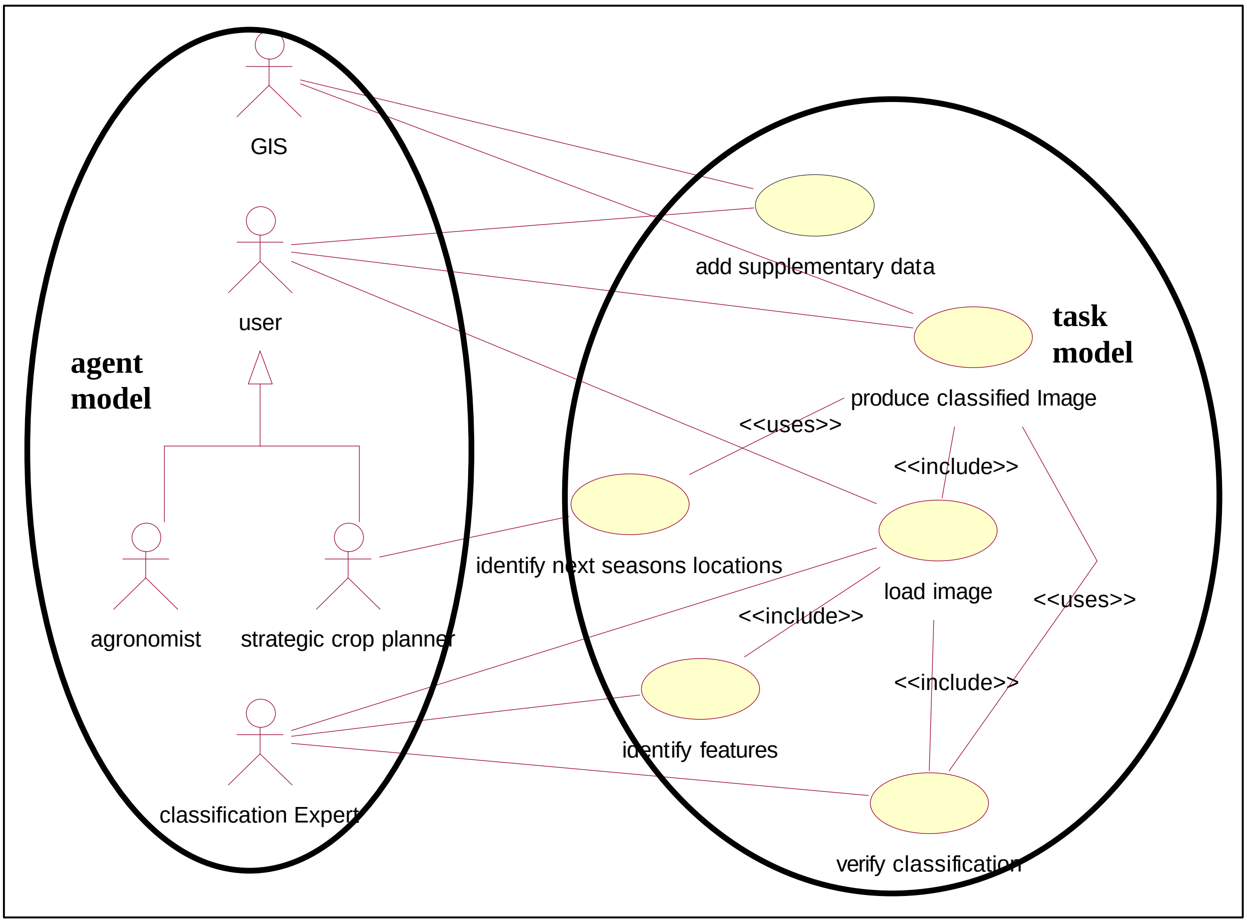


Figure 6 Task model showing relationship to the agent model

3.3. Agent Model

The agent model is fairly straight forward and describes the actors which interact with the overall system. In the case of the agricultural system these are the end users of the system, generally those who want a classified image, an image classification expert, and finally a GIS package which may be useful to manipulate the classified images. Individual end users can have multiple roles and each role is shown as an agent. For example the *strategic crop planner* and the *agronomist* may be the same person, and both roles fall under the more generalised role of *user*. There is interaction between these agents as well, for example the strategic crop planner could use a GIS to view the classified image and next seasons potential crop sites (not all of which may be used).

In this example the agent user has not been fully refined, as the request for classified images has come from individual companies and the specific user roles within those companies has not been identified. Some possibilities are shown here and in the organisational model.

Each agent would have a template similar to below associated with them

NAME Image classifier
INVOLVED IN Load image, identify features, verify results
COMMUNICATES WITH user, GIS
KNOWLEDGE how to identify features on a satellite image, band combinations
OTHER COMPETENCES use of a GIS or image processing software
ORGANIZATION consultant
RESPONSIBILITIES AND CONSTRAINTS

3.4. Knowledge Model

Initially, all the effort in the agricultural crop identification domain was concentrated on the knowledge model, specifically knowledge for producing a classified image. The knowledge model in the agricultural system contains reusable knowledge at all levels. This model is determined by the requirements of the organisation and task models. In a series of related systems, reusable components would be placed in a library. For a specific application those components would then be extracted. If new components were required, they would be created, and then added to the library.

3.4.1. Task knowledge

Task knowledge is not the same as the task model, although there is a relationship In the CommonKADS methodology high level abstract classes would be associated with the task level.

For example a key task in the task model *produce classified image*, which in this case study is a representation of the various crops of interest. The task knowledge necessary to achieve this would be held at the abstract class *image primitive* level rather than the concrete level such as *poppy*. The classification task template would be as follows with a naming convention from UML and stored as a operation in the image_primitive class:

GOAL : Establishing the correct class for an image component
TYPICAL EXAMPLE : Classification of a crop(Image_Primitive.assign_label)
TERMINOLOGY :
OBJECT : image_primitive.polygon.paddock.paddock#174
CLASS :image_primitive
ATTRIBUTE : image_primitive.polygon.pixel_signature_max_val
image_primitive.polygon.pixel_signature_min_val,
image_primitive.band_combinationl,
FEATURE: attribute-value pairs for an object,
eg. "pixel_sig <= pixel_signature_max_val
and pixel_sig>= pixel_signature_min_val
and band_combination = NDVI"
INPUT : multi band satellite image in raster format
OUTPUT : the class instances found

Task knowledge therefore is the knowledge required to fulfil the tasks identified in the task model. Each task in the task model would therefore have at least one equivalent template in the task layer of the knowledge model.

3.4.2. Inference Knowledge

Inference Knowledge is how to apply domain knowledge to meet a particular task. In the agricultural domain there are three main types of feature which require inference. These are:

- image primitives, such as polygons, points and lines,
- features which are determined through their relationships to other features, usually liners or points and
- more complex features which are combination of image primitives.

In figure 7, these have been identified *as Primitive Knowledge*, *Relationship Knowledge* and *Assembly Knowledge* respectively (Crowther 1998).

INFERENCE primitive
ROLES:
INPUT: pixels, bands, algorithm
OUTPUT: classified primitive (label assigned)
STATIC: classification model
SPECIFICATION
Each time the inference is invoked it generates a label for an image primitive in the output image
END INFERENCE

The above template defines the inference for image primitives. This knowledge could be reused in many different domains which require classification of scenes. The detail of how it was classified would be held in the static classification model defined at the domain level. The inference knowledge is therefore used to achieve the task goal of classifying the image.

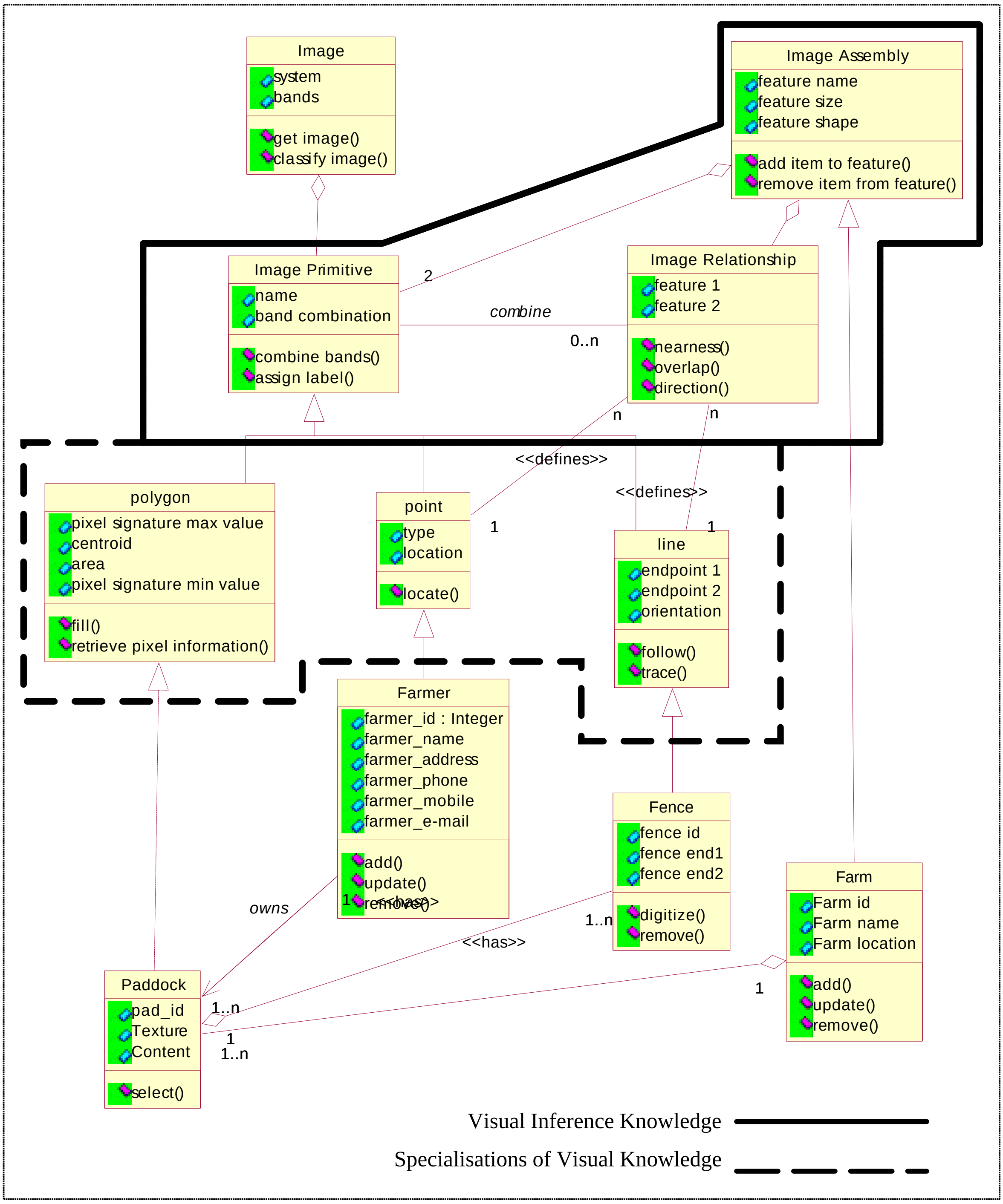


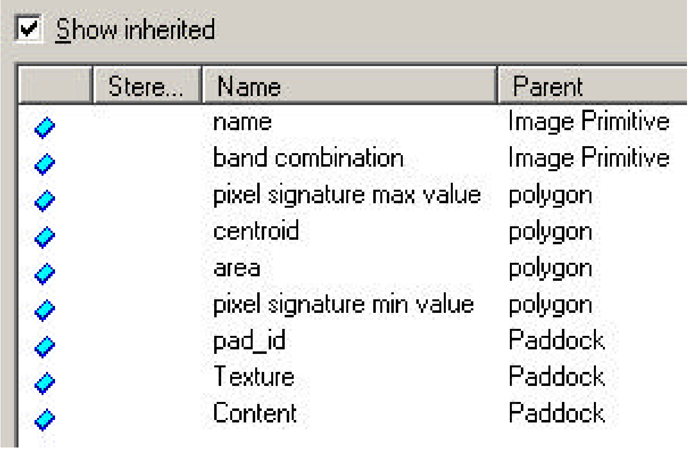
Figure 7 CommonKADS Inference knowledge with examples of associated object classes which would be defined at the domain knowledge level

3.4.3. Domain Knowledge

Knowledge about individual instances of objects in the problem domain would be defined here. This is shown at a high level in figure 4 where the relationships between the various domain classes is described. From the organisation model, the prime objectives of this system is to add a label to each instance of the paddock class. This label would be added by assigning a crop to it.

From a CommonKADS viewpoint, domain knowledge for this system is used required by the inference knowledge which in turn is required by the task knowledge. Figure 7 show the relationship between objects at the inference level and examples of objects at the domain level.

Temporal knowledge (which is not visual) is also held at the domain level. This also relates to image primitives (paddocks) in successive growing seasons. In this example it includes knowledge about crop rotation (for example the same crop is not grown in the same paddock two years in a row unless it is pyrethrum).

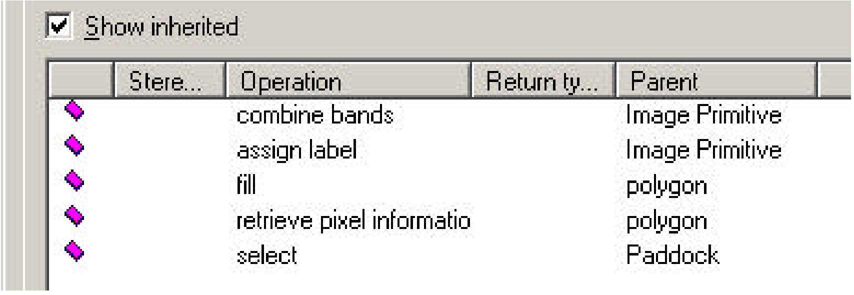


☑ Show inherited

	Stere...	Name	Parent
◆		name	Image Primitive
◆		band combination	Image Primitive
◆		pixel signature max value	polygon
◆		centroid	polygon
◆		area	polygon
◆		pixel signature min value	polygon
◆		pad_id	Paddock
◆		Texture	Paddock
◆		Content	Paddock

Figure 8 UML attribute definition from class *Paddock* showing inherited attributes

Domain knowledge is therefore held in a series of templates about attributes and how those attributes are acquired. Figure 8, taken form the UML domain model shows the attributes which would be filled in for each instance of the object class *paddock*, and figure 9 the associated operations for that class. It should be noted that *paddock* inherits the attributes and operations from classes *polygon* and *image primitive,* as a paddock is a polygon which is a image primitive.



☑ Show inherited

	Stere...	Operation	Return ty...	Parent
◆		combine bands		Image Primitive
◆		assign label		Image Primitive
◆		fill		polygon
◆		retrieve pixel informatio		polygon
◆		select		Paddock

Figure 9 The operations or methods from the *Paddock* entity as defined in UML class operations.

4. Discussion

CommonKADS provides a series of models in which any knowledge based system can be developed. In a general sense it provides the organisational objectives of the system, the organisational setting, the users, both human and system, and the specific goals. The knowledge model provides the knowledge require to meet these goals and objectives

4.1. Spatial Extensions to CommonKADS

There are no direct extensions needed to the CommonKADS model to develop smart or expert spatial systems, however at the knowledge model level it is useful if a template for different types of spatial knowledge is used. The one suggested here is of primitive, relationship and assembly knowledge (Crowther, 1998). This provides an overall framework which can be used for the task and inference layers in the knowledge model.

4.2. Generalised CommonKADS Framework

There are many ways of producing a classified satellite image including the use of neural networks (Openshaw, 1993), knowledge based approach (Crowther et al, 1997) and clustering algorithms (Richards 1993) . The method used in the system described was a knowledge based approach using a pixel level classification, although all fit in the commonKADS framework. In other words, CommonKADS is generalised enough to be used directly in knowledge based spatial applications.

4.3. CommonKADS library

An advantage of building a CommonKADS series of models is that an ontology of the domain is developed. Wielinga and Schreiber (1993) use the following basis for a description of ontologies. A knowledge base can be viewed as a model of some part of the world, that allows for reasoning to take place in that world model, given some inference mechanism. The model is described in a particular language and has a vocabulary and syntax. An ontology defines the constraints of possible objects expressed in the model in addition to the constraints imposed by the syntax.

The ontology developed as part of this project could be reused on similar types of projects, but to do that it would have to be made available as part of a library. The CommonKADS Expertise Modelling Library (Valente, 1998) is an example of such a library, as is Ontoligua server (Farquhar et al 1996). One possible future development is the setting up of a library of spatial ontologies.

5. Conclusions

Many knowledge based systems have been developed, generally with a lot of attention to reasoning strategies, but few seem to have been developed using an overall set of interrelated models. CommonKADS provides these models without specific spatial extensions which can be easily represented using the now widely accepted UML notation. These document the goals and objectives of a system and place it within the organisational context where it will be used.

The advantage of using a series of integrated models to develop knowledge based geographic systems is that not only is knowledge modelled, but also the organisational requirements and the environment it will operate in. This is an aspect that appears to have been lacking in the development process of most spatial knowledge based systems. By using this type of modelling, an information system that meets a specific users needs can be developed in a form that encourages structure and reuse of components, be they knowledge components or interface components.

Given that CommonKADS is designed to exploit ontologies in the form of the CommonKADS Library (Valente et al, 1998), collaborative systems could be built reusing knowledge from the domain layer and inference layer of the knowledge model. In other words there is no need to build specific spatial extensions to CommonKADS, rather a library of model components of a spatial nature could be developed. These components could then be used for rapid development of other systems with a spatial base.

6. Acknowledgements

The work reported in this paper has been supported by a Horticultural Research and Development Grant (Project No. HRDC VG97011).

7. References

Anderson, S. J., Williams, R. N. and Crowther, P. (1999), Automatic detection of Agricultural Field Boundaries Using High-Resolution Visible Light Satellite Images, *Proceedings of the 27th Annual International Conference and Technical Exposition of the Australasian Urban and Regional Information Systems Association,* Fairmont Resort, Blue Mountains NSW.

Barrett, R., Crowther, P., Laurence, R. and Lincolne, R. (2000) 'Agricultural Crop Identification Using Spot and Landsat Images In Tasmania' *,IAPRS, VOL. XXXIII, Amsterdam, 2000.*

Crowther, P., Hartnett, J. and Williams, R. N. (1997), 'A Visual Tool For Capturing The Knowledge Of Expert Image Interpreters: A Picture Is Worth More Than A Thousand Words'. *IGARSS'97,* Singapore.

Crowther, P., 1998, Knowledge Acquisition from Satellite Images for Geographic Expert Systems – Show or Tell?, *Proceedings of the International Conference on Modeling Geographic and Environmental Systems with Geographic Information Systems*, Hong Kong, June

De Hoog, R., Benus, B., Metselaar, C., Vogler, M., and Menezes, W., 1994, *Organisation model: Model definition document.* Deliverable DM6.2c, ESPRIT Project P5248 KADS-II/M6/M/UvA/041/3.0, University of Amsterdam and Cap Programmator.

Farquhar, A., Fikes, R. and Rice, J. 1996. *The Ontolingua Server: a Tool for Collaborative Ontology Construction*. Standford Knowledge Systems Laboratory. [Online] Available http://www.ksl.stanford.edu/software/ontolingua/project-papers.html (last accessed 8th July 2003)

Fensel, D. and Van Harmelen, F., 1994, A Comparison of Languages which Operationalize and Formalise KADS Models of Expertise, *Knowledge Engineering Review*, Vol. 9, No. 2, pp.105 - 146.

Gustafsson M. and Menezes, W., 1994, *CommonKADS Reference Manual*, Esprit Project P5248 KADS-II, KADS-II/P2/WP/CP/001/1.0, 05/1994

Motta, E., 1997, Trends in Knowledge Modeling: Report on the 7th KEML Workshop. *The Knowledge Engineering Review*, V12(2), pp. 209-21

Openshaw, S., 1993, Modelling Spatial Interaction Using a Neural Net, *GIS Spatial Modelling and Policy Evaluation*, Springer-Verlag, pp. 147 - 164.

Richards, J. A., 1993, *Remote Sensing Digital Image Analysis*, Springer-Verlag, New York, Chs. 4, 6 and 8.

Rumbaugh, J., Jacobson, I. and Booch, G., 1999, *The Unified Modelling Language Reference Manual*, Addison Wesley.

Schreiber, G., Wielinga, B. and Breuker, J., 1993, *KADS - A Principled Approach to Knowledge-Based System Development*, Academic Press, San Diego

Schreiber, G., Wielinga, B., Akkermans, H. Van de Velde, W. and Anjewierden, A., 1994, CML: The CommonKADS Conceptual Modelling Language, *Proceedings of the 8th European Knowledge Acquisition Workshop*, Hoegaarden, Belgium.

Schreiber, G., Akkermanns, H., Anjewierden, A., De Hoog, R., Shadbolt, N., Van de Velde, W. and Wielinga, B., 2000, *Knowledge Engineering and Management: The CommonKADS Methodology*, MIT Press.

Valente, A, Breuker, J. and Van de Velde, W., (1998), The *Common*KADS library in perspective, *International Journal of Human-Computer Studies*, V49, pp391-416.

Wielinga, B. J., Schreiber, A. T. and Breuker, J. A., 1992, KADS: A Modelling Approach to Knowledge Engineering, *Knowledge Acquisition*, Vol. 4, No. 1.

Wielinga, B. J. and Schreiber, A. 1993. Reusable and Sharable Knowledge Bases: A European Perspective. *Proceedings of the International Conference on Building and Sharing of Very Large-Scaled Knowledge Bases '93.*