

Configuration Irrigation schedule Based on Expert Systems and Operations Research

Iman Hassen^{*}, Mohamed Rasmy^{**}, Ahmed Rafea^{***}

^{*}Central Lab for Agriculture Expert System, ARC

Email: iman @ mail.claes.sci.eg

^{**}Faculty of Computer and Information, Cairo University

Email: m_h_rasmy @ hotmail.com

^{***}Computer Science Dept., AUC

Email: rafea@aucegypt.edu

Abstract: This paper presents configuration design knowledge to solve irrigation scheduling problem in agriculture domain and a general approach for configuration management knowledge. It also introduces a Configuration Agriculture Irrigation Schedule (CAIS) tool that produces the configuration irrigation schedule design according to the CommonKADS methodology. A configuration that satisfy the user requirements, and that additionally satisfy some optimality criteria is required. By integrating Dynamic programming and inference knowledge, with demand including different objective, that it is used in generating the inference steps (PPSM) to select the suitable components of the inference steps minimizing the number of components and/or maximizing the accuracy of the result of each inference step. Operations Research techniques greatly improved the understanding of the interactions of a large number of components and enhanced the decision selection under uncertain outcomes.
Copyright © 2004 IFAC

Keywords: Configuration, Operations research, Expert systems, Knowledge-based systems, Knowledge acquisition, Artificial intelligence, Dynamic programming.

1. INTRODUCTION

Recent approaches integrate both the Operations research (OR) and Expert Systems (ES) together under a unify framework. Although the benefits of applying OR and ES technology can be significant, developing real world extremely time consuming task. Modeling an agriculture domain requires incremental modifications to candidate schedules need to be designed and implemented to generate a configuration design includes these two approaches.

Configuration design is a form of design where a set of pre-defined components is given and an assembly of selected components is sought that satisfies a set of requirements and obeys a set of constraints (Mittal, S., 1989 & Yu, B,1995). Sometimes an optimality criterion may be given that defines an ordering upon possible solutions. Following Chandrasekaran (1990) one could argue that a specification of the technology to be applied in the design process is also part of the input.

In addition to the problem specification—knowledge is needed about the design domain (domain theory), additional requirements and/or constraints that are not explicitly given as input but that are implicitly given in regulations, common sense etc., optimality criteria and optionally certain preferences. Preferences are design choices which are not the outcome of the design process itself, but which are

given beforehand and constrain the space of design options. Such design choices may be changed when no solution can be found within the constrained search space. We propose Configuration Agriculture Irrigation Schedule (CAIS) tool according to the CommonKADS methodology (Wielinga, 1994) which could overcome some of the limitation found in previous research efforts

In the following section, a Configuration Agriculture Irrigation Schedule (CAIS) components are described, in section three the processes of CAIS tool are presented, the conclusion is in section four.

2. CONFIGURATION AGRICULTURE IRRIGATION SCHEDULE (CAIS) COMPONENTS

A Configuration Agriculture Irrigation Schedule (CAIS) tool consist of three main components, namely Agriculture Schedule ontology, OR library, and expert systems library. This tool produce one intermediate result called 'irrigation knowledge base' through the Knowledge Acquisition process, and it generate a configuration irrigation schedule design through the generation process as depicted in figure 1. The following subsections describe each component of the three main components in addition to the intermediate result and the final product of the tool.

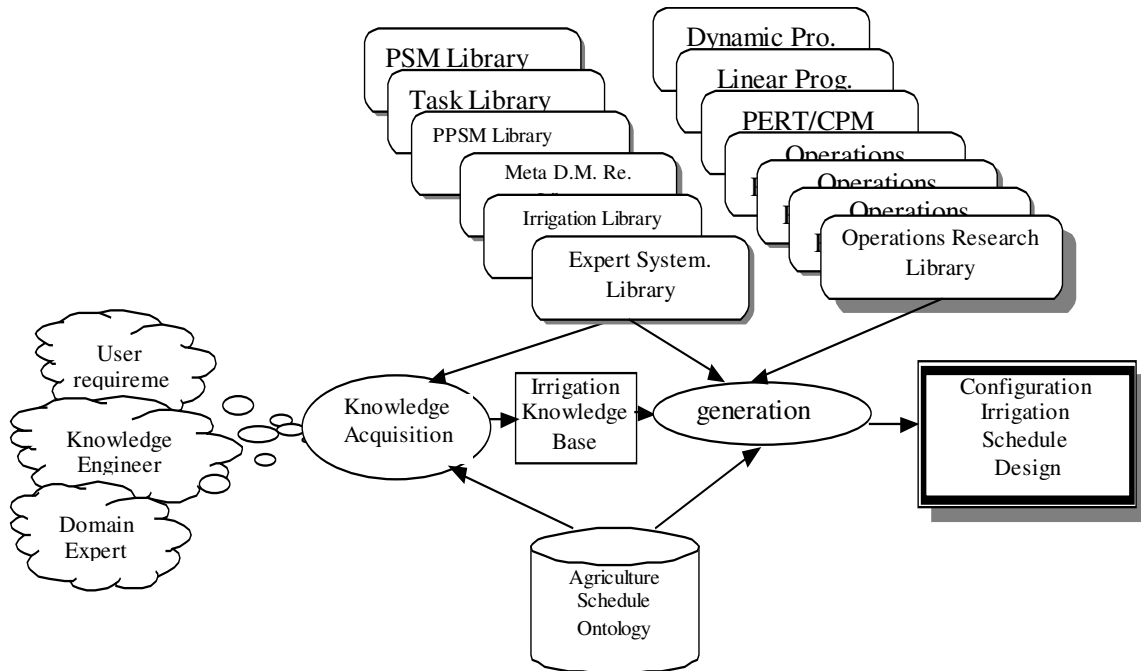


Figure 1: Generating Configuration Irrigation Schedule Design

2.1 AGRICULTURE SCHEDULE ONTOLOGY

An ontology for a knowledge based system is an explicit specification for the objects, concepts and other entities that are presumed to exist in some area of interest as well as the relationships that hold among them (Gruber T.R., 1993).

In CAIS Agriculture schedule ontology classified into five parts: resources, irrigation domain ontology (activity), irrigation schedule (product), user requirement (demand), and constraint.

Resources are an entity that supports or enables the execution of activities (Stephen F. 1997). Resources are generally in finite supply and their availability constraints when and how activities execute. In CAIS, we conceptualize the resource as consisting of objects (i.e. soil, water, climate, farm, previous crop, and plantation) that are elementary entities such as integers and string (i.e. soil salinity, soil texture).

The activity is a set of processing steps required to produce or provide the product (irrigation schedule). It is the set of activities that fulfill the demand. The Sequences of activity are expansion, Et0, EtCrop, SHWC, Water Requirement, and Frequency/Interval to achieve the user requirement. An irrigation domain ontology (domain model) defines declaratively the set of terms and relations of a domain independent of any problem solving method. The domain model contains static information about the application environment.

2.2 OPERATIONS RESEARCHSH LIBRARY

Operations Research library is a collection of algorithms on dynamic programming, linear programming, and pert/cpm techniques that used in irrigation schedule. The dynamic programming is used in generating the inference steps (PPSM) to select the suitable components of the inference steps minimizing the number of components and/or maximizing the accuracy of the result of each inference step. A dynamic programming formulation that used in generating inference steps described as follows:

$$f_t(s_t) = \max/\min (r(d_t, s_t) + [f_t (s_{t+1})])$$

$$\text{Subject To}^{d_t}$$

$$S_{t+1} = g_t (d_t, s_t)$$

$$S_t \in S_t, d_t \in D_t, t = 1, 2, 3, \dots, n$$

Where $f_t(s_t)$ is the maximum expected total return $r(d_t, s_t)$, $t=1,2,\dots,n$ finite stages of tasks, s_t is a finite state of a Primitive Problem Solving Method, d_t is a decision for each task, The state of the system in $t+1$ is described by the generation function $g_t (d_t, s_t)$. The objective function can be changed to minimize the components as follows: $f_t(s_t) =$ minimum component of PPSM (inference step) when it is in PPSM (state) s with n more tasks (stages) to generate the inference step.

Another dynamic programming formulation that used in a dialy irrigation task described as follows:

$$f_t(s_t) = \max_{d=0,1} [(Y_t * P_t - (C_w * q_t * I_t + I_t * C_E * C_L) + f_t (s_{t+1})]$$

$$t=1,2,3,4 \dots \dots \dots T$$

Subject To

$$O_t = Et0(D, T, R) \tag{1}$$

$$E_t = EtCrop(O) \tag{2}$$

$$S_t = SHWC \quad (3)$$

$$W_t = E_t + E_{t-1} \quad (4)$$

$$S_t + R_t - W_t = q_t \quad (5)$$

$$E_0 = 0 \quad (6)$$

$$I_t = \begin{cases} 0 & q_t > 0 \\ 1 & \text{and } E_t = 0 \quad q_t \leq 0, \end{cases} \quad (7)$$

Where $f_t(s_t)$ is the maximum profit of expected yield in ton, $t=1,2,\dots,n$ finite stages of plant age per day, Y_t is the expected yield at certain stages, P_t is the price per ton, C_w is water cost; w_t is the water requirement, I_t is an irrigate variable taking the value 1 for irrigate or 0 if not., C_E represents Energy costs for each irrigation, C_L denote labor costs. O_t is a value of E_{t0} inference step that determined by current date, daily temperature and relative humidity that denoted by D , T , R respectively. E_t is a value of E_{tCrop} inference step that determined by plant age and E_{t0} that denoted by t and o_t respectively. S_t is a value of $SHWC$ soil available water inference step that determined by plant age. R is a rain, and q_t is a dummy variables to *take decision* in $\{0, 1\}$, 1 means start irrigation today, 0 means wait until to next day. The whole description of those formulations will be explained in section 3.

2.3 EXPERT SYSTEMS LIBRARY

Expert system library is a collection of problem solving method (PSM), Tasks, primitive problem solving method (PPSM), Meta domain knowledge representation, and irrigation libraries. The roles that need to be filled by a PSM prescribe what domain knowledge is expected from the expert. The domain knowledge is, therefore, no longer something that is acquired and represented independently of how it will be used in concrete problem solving. However, it is still explicitly and separately represented, giving the advantage of maintainability and systematic ness in knowledge acquisition. This library contains nine PSMs irrigation schedule

The task reflects the general problem solving method for scheduling. A problem solving method can be defined as a knowledge-use-level characterization of how a task might be solved. It can also be seen as an abstract model of how to solve certain problem. In

```

task: Irrigation schedule ,
task-definition:
  goal: the main goal of the irrigation is to determine the water requirement during
  cultivation.
  input: case-description
  output: irrigation schedule
task_body
  type: Composite
  sub_tasks: Propose mathematic irrigation schedule
  control_structure:
    Propose mathematic irrigation schedule.
task: Propose mathematic irrigation schedule
task-definition:
  goal: Generating an propose irrigation schedule
  input: case-description
  output: Propose irrigation schedule
task_body
  type: Composite
  sub_tasks: Get case description , determine plant parameters,
    Initialize irrigation parameters, Compute propose
    irrigation schedule, irrigation units,
    Display irrigation schedule
  control_structure:
    Get case description ,
    case description, expansion model -> expanded case description,
    Initialize irrigation parameters,
    Compute propose irrigation schedule,
    water requirement, unit model -> irrigation unit,
    Display irrigation schedule.

```

Figure 2: task knowledge of irrigation

irrigation schedule design, a generic problem solving method called “propose and revise” is based on mathematics model and heuristic model. Irrigation Task library contains 18 tasks for irrigation schedule for crop. Irrigation is the main task of the irrigation generates design. The goal of the irrigation task is to get irrigation schedule of the case description. Its inputs are the case description. Its output is the irrigation schedule of the case description.

2.4 IRRIGATION KNOWLEDGE BASE

The irrigation knowledge base is the result of the knowledge acquisition process. It contains all the related knowledge for the crop, Evapotranspiration (E_{t0}) method used, E_{tCrop} methods used, Available water in soil methods used, and water requirement methods used.

2.5 IRRIGATION SCHEDULE DESIGN

Irrigation schedule design is the final product to satisfy the user requirement. There three main components of the irrigation schedule design: task knowledge, inference knowledge, and domain knowledge as described in the following subsections.

2.5.1 IRRIGATION TASK KNOWLEDGE

The design of task knowledge consists of two parts: the task definition and the task body. The task definition describes the main system goal, as well as the input and the output roles. The task body describes the task type (composite or primitive), subtasks (if any), additional roles (which are not included in either input roles or output roles), and the control over these subtasks. Figure 2 shows a generated part of task knowledge of irrigation. It shows that the irrigation task contains only one task, namely: 'Propose mathematic irrigation schedule'. The input is case-description and the output is proposing irrigation schedule. Type of task is composite. The irrigation schedule subtasks involves two Transfer Tasks, one Primitive Task, and one composite task namely: 'Get case description', 'Display irrigation schedule', 'Initialize irrigation parameters', and 'Compute propose irrigation schedule' respectively. There are also two Task Static Primitive (inference step) are expand and irrigation units.

Name	:Expand
Goal	:Derive parameters used in the system according to the available data involved in the case description role
Static Role	: Expansion model
Input Role	:Case description
Output Role	:Expanded case description
Spec	:soil.type : relation Plant age : function

Figure 3: inference knowledge of irrigation

2.5.2 INFERENCE KNOWLEDGE

The inference describes how domain knowledge is used in elementary reasoning steps (or inferences). The inference knowledge is described using two views namely: inference structure and inference specification (Breuker J. et al., 1994). The design of inference knowledge consists of two main parts namely: inference structure and inference specification. An inference structure diagram is used to describe this type of knowledge. Three types of components are used in this diagram, namely: inference steps, dynamic roles, and static roles. An Inference specification contains five parameters: name, static role, input role, output role, and specification. A name of inferences represent the role that these inferences play in solving the problem, static roles indicate which domain model should be accessed, input and output roles are supposed to be a part of the overall working memory of the problem solver and are thus not directly linked to specific domain model. The first inference expand (see figure 3) uses the expansion model to derive a new data using the available data included in case description role.

2.5.3 DOMAIN KNOWLEDGE

The domain knowledge layer consists of two parts: the set of ontology related to the application domain and domain models. Domain ontology is considered as the most fundamental constituent of domain knowledge. It defines the language of the domain, i.e., the terms used to describe the domain. This language consists of a number of constructs such as concepts, expressions, and relations (Gruber T.R., 1992). The word "ontology" can be defined as "what exist in the real life application". Generally speaking, the scheduling ontology should be mapped to five classes of concepts: resources, irrigation domain ontology (activity), irrigation schedule (product), user requirement (demand), and constraint.

A domain model is defined as a coherent collection of expressions about a domain that represents a particular viewpoint defined in ontology. The domain model may therefore embody certain assumptions

that are specific for the ontology that is used. Domain model have an internal structure for knowledge representation.

Irrigation schedule domain model includes 9 domain models. The irrigation domain model are expansion model, Evapotranspiration(Et0) model, EtCrop Model, Water Requirement model, SHWC model, Frequency_interval_model, and Constraint Model. Each domain has its classification (i.e Evapotranspiration (Et0) model classify into Et0 Penman method, Eto modified Penman method, Et0 Penman Mountance method, Et0 Hargrve method, ET0 Blaney method, ET0 Radiation method, Et0 Penman Monteith method, ET0 Pan evaporation method, Et0 Reference based on Sector, Eto Reference based on Governor, and Et0 Reference based on Directora.). Each domain model classification contains a parts which describe the knowledge representation for this method. For example, Et0 Hargrve method consists of four parts: 1)Eto Hargrve for current month, 2) Eto Hargrve for next month, 3) Smooth factor, and 4) adaptive Eto Hargrve

3. CONFIGURATION AGRICULTURE IRRIGATION SCHEDULE (CAIS) PROCESSES

There are two main processes in the CAIS tool, the first one is a knowledge acquisition, and the second is generation as shown in figure 1. The first process produces the irrigation knowledge base, and the second one produce the irrigation schedule design. The next two subsections describe those processes.

3.1 KNOWLEDGE ACQUISITION

The irrigation knowledge base is the result of the knowledge acquisition process. In the knowledge acquisition process, the knowledge engineer collecting the user requirement and eliciting the knowledge using the agriculture schedule ontology and the expert system library by using a special knowledge acquisition tool that consists of five parts crop parameters, Evapotranspiration (Et0) method, EtCrop methods, Available water in soil methods, and water requirement methods. After the knowledge engineer collecting the knowledge relate to crops, the domain expert uses this tool to verify the crop parameters.

3.2 GENERATION

The generation starts with the requirements definition where several questions are used to guide the generations of the irrigation schedule design for any crop. The criteria for generating design describe by the property (Attribute) and the configuration design for irrigation system is used to collect data on:

- Generation model (e.g. mathematical, heuristic...)

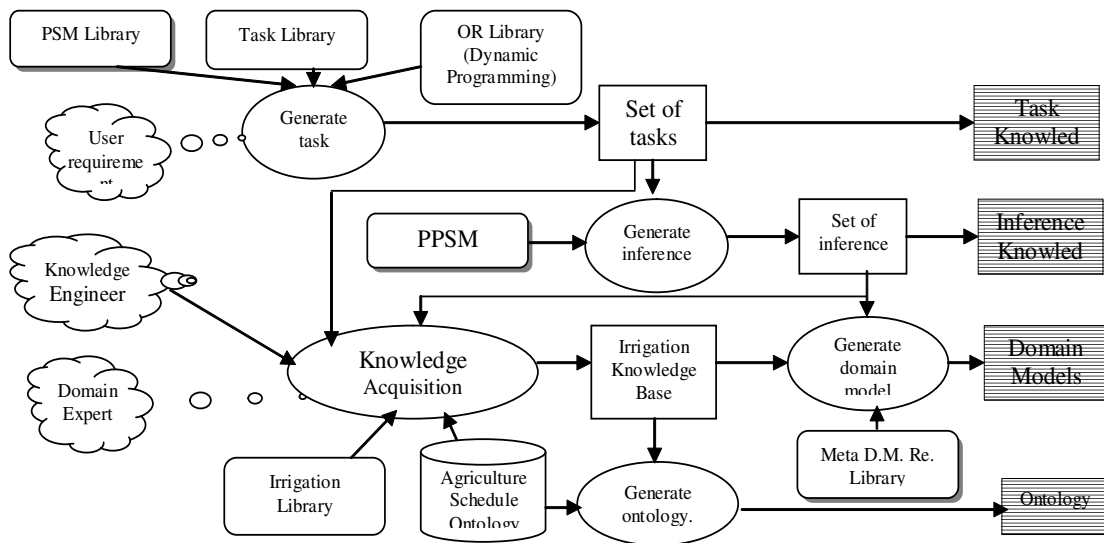


Figure 4: Detail Process of Generating Configuration Irrigation Schedule Design

- Schedule type(daily, weekly, each-10-day, monthly)
- The objective of irrigation management (i.e. maximize net return, minimize irrigation costs, maximize yield, optimally distribute a limited water supply)
- Scheduling problem solving containing propose and revise (i.e., propose, revise)
- Events for reactive scheduling (i.e., forest, wind, rain...)
- Disease (i.e., fungal...)
- Yield response to water (i.e. forecasting expected yield)
- Minimize the component of design (yes, no)
- Evaporation method (Et0 Penman, Eto modified Penman, Et0 Penman Mountance, Et0 Hargrve, ET0 Blaney, ET0 Radiation, ET0 Pan evaporation, Et0 Reference Sector, Eto Reference Governor)

The process generation is decomposed into four processes namely: generate tasks, generate inference, generate domain model, and generate ontology to produce the irrigation schedule design that contains four parts: tasks, inference, domain model, and ontology as shown in figure 4. The next subsections describe each one of them.

3.2.1 GENERATE TASK

Irrigation schedule task has sub-tasks. The set of sub-tasks of a specific method have some sort of part-of relationship with the method. Each sub-task might itself be realized by different methods. Thus, the task-method decomposition is recursive until primitive inferences (or primitive tasks in our terminology) are reached, calling the mechanism of generating task by the "irrigation task schedule". There are four type of tasks: Primitive, Transfer,

static Primitive, and composite task. In case of composite task, the generate task mechanism calls itself recursively to get the sub of problem solving method or primitive problem solving method. The irrigation scedule task contains two PSMs are 'Propose and revise irrigation schedule mathematical model' and 'Propose and revise irrigation schedule heuristic method'. It must select only one PSM based on the user requirement.

The mechanism generate a set of task using PSM library, task library, OR library, and user requirement. Consultring the task library and match each selected task stored in the dynamic list of generated tasks to write its knowledge in the task knowledge.

3.2.2 GENERATE INFERENCE

By using one of the OR library called dynamic

```

Function domain_model_generation.
Begin
  get List_of_inference from inference_DB(PPSM)
  While(not empty List_of_inference)
  Begin
    get name_PPSM from List_of_inference
    get static_role of name_PPSM
    store static_role in domain_model_DB(Domain_Model)
  End/ While ]
  Get production_schedule.system = System
  get List_of_DM from domain_model_DB(Domain_Model, System)
  While(not empty List_of_DM)
  Begin
    get Domain_model from List_of_DM
    get System from List_of_DM
    getDomain_model_representation(System,Domain_model,_, Representation,
    Case Representation
    Relation:
      Procedure generate_relation_DM(table name(Item), input_output, position text,
    text)
    Table:
      Procedure enerate_table_DM(Domain_model, System)
    Function:
      Procedure generate_function_DM( Domain_model, System)
    End(Case)
  End( While )

```

Reference based on Sector Eto Database based on G... Directorate.]]. The decision at each task (stage) is

denoted by D_t . When the process comes to PPSM (states) of some task, it can make different decision to determine the PPSM of the next task. The value of $D_t(S_t)$ is used to denote the decision made of PPSM S_t of task T . The state generation inference knowledge of the CAIS system in $t+1$ is described by the function $gt(D_t, S_t)$ and the return in each period $rt(dt, st)$. The objective function is dynamic according to the user requirement. (i.e. Minimize the component of PPSM result; maximize the accuracy of PPSM result,...). Each PPSM (state) has the path (arcs) to represent the return value of the path and next PPSM (state). We add for each PPSM (state) the knowledge base concerning with each return value based on the objective function. Constraints are the most important part of every scheduling problem. There are two type of constraint. First, a set of constraints on the value of resources. Second, constraint model, a set of constraint on the propose irrigation schedule according to objective function to adaptive the irrigation schedule to satisfy on of the following objective maximize net return, minimize irrigation costs, maximize yield and optimally distribute a limited water supply.

3.2.3 GENERATE DOMAIN MODEL

Generate domain model uses irrigation knowledge base, meta domain model representation, irrigation library, and set of inference as input and produces the domain model. Figure 5 depict the algorithm that used to generate the domain model.

4. CONCLUSION

Integrating Operations research and Expert Systems under a unify framework in agriculture domain is very important task specially in scheduling system like irrigation system to generate a configuration design includes these two approaches. This paper presents configuration design knowledge to solve irrigation scheduling problem in agriculture domain and a general approach for configuration management knowledge. We present a Configuration Agriculture Irrigation Schedule (CAIS) tool that produces the configuration irrigation schedule design according to the CommonKADS methodology. This configuration satisfies the user requirements, and some optimal required criteria. By integrating Dynamic programming and inference knowledge, including different objective, to select the suitable components and minimize the number of components and maximize the accuracy of the result.

5. REFERENCES

Breuker, J., and Van de Velde, W. (1994). CommonKADS Library for Expertise Modeling: Reusable Problem Solving Components. IOS-press, Amsterdam, August 1994.

Chandrasekaran, B. (1990). Design problem solving: A task analysis. *AI Magazine*, Winter issue, pages 59–71.

Gruber T.R., (1993). A translation approach to portable ontology specifications, *Knowledge Acquisition*. 5 199–220.

Gruber T.R. (1992). Ontolingua: a mechanism to support portable ontologies. Technical Report KSL 91-66 (revision), Knowledge Systems Laboratory, Stanford University, March 1992.

Mittal, S. & Frayman, F. (1989). Towards a generic model of configuration tasks. In *Proceedings of the 11th IJCAI*, pages 1395–1401, San Mateo, CA, Morgan Kaufman.

Smith, S.F., O. Lassila and M. Becker (1996). A Scheduling Ontology: Informal Concept Descriptions , ICLL Working Paper, January, 1996.

Stephen F. Smith and Marcel Becker (1997). An Ontology for Constructing Scheduling Systems, *AAAI Spring Symposium on Ontological Engineering*, Stanford, CA, March, 1997 (AAAI Press).

Yu, B. and MacCallum, K. (1995). Modelling of Product Configuration Design and Management by Using Product Structure Knowledge. *Int. Workshop on Knowledge Intensive CAD*, Finland, 1995.

Wielinga, Bob J. (1994) Expertise Model Definition Document, ESPRIT Project P5248 KADS-II, Document Id.: KADS-II/M2/UvA/026/5.0, University of Amsterdam, 1994.