Using Formal Metadata Descriptions for Automated Ecological Modeling

Virgínia Brilhante

School of Artificial Intelligence - Division of Informatics - University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN, UK
virginia@dai.ed.ac.uk

Abstract

System dynamics is a mathematical modeling approach widely used in environmental studies as a tool for representing and simulating ecological systems, giving support to prediction and decision making. The knowledge sources for model design are, essentially, ecological data and human expertise. Interestingly, however, the direct influence of data properties on model design has been little explored. We hypothesise that property descriptions of ecological data (ecological metadata), such as functional, temporal and spatial relations between variables, can be used to guide and to substantiate structural modeling decisions. In this paper we address the use of formal descriptions as an approach to representing ecological metadata, enabling us to automatically draw links between these formal descriptions and ecological modeling. A working example is presented in detail to illustrate the approach.

Linking Metadata to System Dynamics Model Design

Systems dynamics, in particular, is the ecological modeling (Gillman & Hails 1997) paradigm whose automation we are investigating. A system dynamics model represents a system by means of compartments, flows, influence factors and influence links. Compartments correspond to stocks of material or energy in the ecological system being modeled. The amount 'in stock' depends on in-flows, which increase the amount, and out-flows, which decrease the amount. For instance, the compartment *biomass* can have as an in-flow *vegetation growth* and as an out-flow *litter production.* Flows can occur between compartments, from the outside (i.e. from somewhere in the ecological system beyond the model's scope) to a compartment, or from a compartment to the outside. Each compartment has an associated state variable whose value represents the amount of material or energy in the compartment. Running a model consists of calculating the changes in the values of the state variables, given initial conditions and mathematical equations governing the flows. Such mathematical equations express how the flows are affected by compartments and other influence factors linked to it. E.g., *vegetation growth* (flow) can be affected by *biomass* (compartment) and *rainfall* (influence factor). Complex nets of influence links can be represented in a system dynamics model.

Designing a system dynamics model is not a one-off task. Rather, it is an incremental process, comprising various kinds of decisions and developments that can be iteratively refined. A simplified outline of one iteration of a typical system dynamics modeling process is given below:

1. *Model Purpose*: The first decision taken, and one that will influence all the others, is the model purpose: the precisely defined question that the model is expected to answer once built.
2. *Model Structure*: Next, the modeler represents the ecological system at issue as interconnected compartments, flows, influence factors and influence links. Here, the model structure is designed and the functional relationships between model elements (represented by influence links) are determined.
3. *Equation Design*: The structure above is expressed mathematically. Each model element is defined as a mathematical equation.
4. *Derivation of parameters and estimation of initial values*: Parameters in the equations above are derived and initial values to the compartment variables are estimated based on a dataset and other sources of information.

Complete automation of this process is impractical, given the state of the art in automated modeling research. Our initial focus is on *derivation of parameters.* This particular task has been chosen as a starting point due to its smaller degree of subjectivity. We understand better (for the time being) how parameters can be derived than, for instance, the reasoning behind *model structuring.* So far we have had early results on automating two parameters derivation tasks. One, given declarative facts expressing structural (e.g. *branch weight of individual trees are measured*), temporal (e.g. *measurements are taken in an annual basis*), and spatial (e.g. *measurements are taken from each site* *in the experimental field*) properties of a dataset and common ecological functions (e.g. *mean*, *max*, etc.), the system derives an exploration of the space of parameters supported by the dataset properties (e.g. *the max branch weight for each individual, each site and each year*; *the year mean of the mean branch weight for each individual and each site*, etc.). Two, given a parameter to be derived, the system generates a progressive sequence of intermediate derivation steps, starting from the descriptions of the dataset properties. The example that appears later in this paper is concerned with this task.

Model structuring, usually the starting point of the mental process carried out by human modelers, should be the following focus. The degree of automation envisaged here comprises having the system indicating candidate model elements and relationships to compose model structure, again based on metadata descriptions that characterise a certain dataset. Next, we plan moving on to *equation design*, where we envisage automatic inference of potential interdependencies between variables.

Formalisation of Ecological Metadata

Descriptions of properties of actual datasets (metadata), accompanied with some extent of modeling knowledge, compose the information substratum we explore in order to partially automate and provide guidance on the design of models which are appropriate to those datasets. This can only be attractive if the mechanisms supporting it are not tied to specific datasets. Pre-defining a detailed knowledge representation system still general enough to express properties of every ecological dataset is infeasible. Thus, what is needed is a general framework for metadata description which can be instantiated to specific datasets and model purposes. We have a prototype ontology, named *Ecolingua*, for such a framework, providing a vocabulary and axioms for ecological metadata description.

The Ecolingua Ontology

An ontology is a shared understanding of some domain of interest, specified in the form of definitions of representational vocabulary. Axioms can be defined to constraint interpretations over the ontology's vocabulary (Uschold & Gruninger 1996). In a broader sense, and beyond the scope of this research, an ontology to express ecological metadata such as *Ecolingua* can act as an inter-lingua for knowledge sharing. People involved with collection, organisation and analysis of ecological data, designers of field sampling strategies and modelers are all potential users.

As initial resources for Ecolingua's design we took, on one hand, an informal methodology found in (Uschold & Gruninger 1996), and on the other, a very diverse dataset generated by a tropical forest logging experiment in the Amazon, Brazil (Biot 1995). First, from the dataset specifications, we abstracted its meta-level structural, spatial and temporal properties. At this stage we started using the Ontolingua Server, our third resource, through which we described those ecological meta-level properties by means of frame-system structures, the foundational theory underlying the architecture of ontologies designed through the server. The Ontolingua Server (Knowledge Systems Laboratory, Stanford University, http://www.stanford.edu) (Farquhar, Fikes, & Rice 1996) is one of the few available tools to date for ontology construction and sharing. One of the main resources provided is an extensive library of sharable ontologies whose definitions can be reused for the development of new ontologies.

To illustrate the ontology and to give examples of ecological metadata, let us present a small excerpt from *Ecolingua* in figure 1.

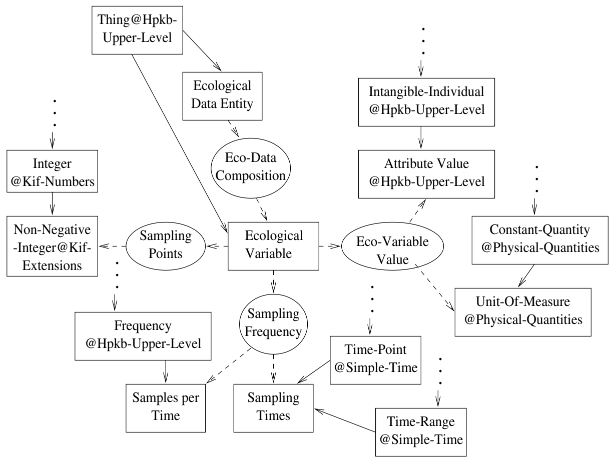

Figure 1: Excerpt from *Ecolingua*

Figure 1 shows part of *Ecolingua* depicted as a hierarchy of classes and relations between them. Boxes represent classes, with directed arcs (full lines) representing class/subclass hierarchy; e.g. *Ecological Data Entity* is a subclass of *Thing@Hpkb-Upper-Level*. The notation *Class-name@Ontology-name* is used for classes defined in other ontologies rather than *Ecolingua*, which are available in the ontologies library of the Ontolingua server; e.g. *Thing* is defined in the *Hpkb-Upper-Level* ontology, *Unit-Of-Measure* is defined in the *Physical-Quantities* ontology. Ellipses represent relations between classes. The directed arcs (dashed lines) show the direction of the relation, from domain classes to range classes; e.g. the relation *Sampling Points* gives a *Non-Negative-Integer* (range) number of sampling points of an *Ecological Variable* (domain).

Ontolingua, the representation language used by the server, was created as an attempt to solve the portability problem for ontologies. It adopts a translation approach in which "ontologies are specified in a standard, system-independent form and translated into specific representation languages" (Gruber 1993). Our target language is Prolog, which we use to connect metadata descriptions to endorsements of model structure. The server automatically translates *Ecolingua* plus all the ontologies it refers to, into a file containing a very large (5.3Mb) ill-structured knowledge base in a Kif-like Prolog-readable version of the Ontolingua syntax. To extract a manageable knowledge base from this file we built tools for syntactic correction, consistency checking, pruning and mapping of logical sentences into more elegantly constructed Horn clauses. The target knowledge base contains vocabulary and axioms that belong to three levels of abstraction: *(1)* definitions specific to ecological metadata description (*Ecolingua*'s definitions); *(2)* definitions inherited from other ontologies through *Ecolingua*'s references to them; and *(3)* definitions related to the meta-ontological frame-based primitive terms (e.g. *class*, *instance*, *slot*, *relation*, etc.), imposed by the server. This is the knowledge base which will form the basis for connections drawing between ecological metadata and model design.

An Example

For illustration of automatic inference of modeling elements based on metadata (*metadata↔model*, for short), we present here a simple example regarding the derivation of model parameters. This is a working example implemented in Prolog.

Model Scope:

The scope for our example is a tropical forest logging experiment where a system dynamics ecological model is designed in order to assist on the prediction of timber production and ecological impact (Biot *et al.* 1996).

Parameter to be derived:

Let us suppose that in one of the equations in the model expressing timber production, a parameter is required to quantify the *average annual increment of dbh*[1] *of trees*.

Dataset properties:

Baseline data of the trees is made available to the logging experiment. A previous logging took place on the same forest plots in 1987 and the DBH of 360 trees has been measured once in 1990 and once in 1995. These data properties can be expressed through *Ecolingua*'s classes and relations in figure 1. Figure 2 shows part of *Ecolingua* as graphically presented in figure 1 in first order predicate calculus.

Inferences:

The drawing of *metadata↔model* inferences based on the properties of a particular dataset, requires *Ecolingua* to be instantiated, i.e., the metadata of the ecological dataset under consideration needs to be described using the vocabulary provided by *Ecolingua*.

The instances of the ontological classes necessary for description of the example's dataset are represented by the facts shown in figure 3.

[1]DBH - Diameter at Breast Height

$class(ecological\_data\_entity).$
$class(ecological\_variable).$

$relation(eco\_data\_composition).$
$domain(eco\_data\_composition, ecological\_data\_entity).$
$range(eco\_data\_composition, ecological\_variable).$

$class(attribute\_value).$
$class(unit\_of\_measure).$

$relation(eco\_variable\_value).$
$domain(eco\_variable\_value, ecological\_variable).$
$range(eco\_variable\_value, attribute\_value).$
$range(eco\_variable\_value, unit\_of\_measure).$

$class(non\_negative\_integer).$

$relation(sampling\_points).$
$domain(sampling\_points, ecological\_variable).$
$range(sampling\_points, non\_negative\_integer).$

$class(samples\_per\_time).$
$class(sampling\_times).$

$relation(sampling\_frequency).$
$domain(sampling\_frequency, ecological\_variable).$
$range(sampling\_frequency, samples\_per\_time).$
$range(sampling\_frequency, sampling\_times).$

Figure 2: Excerpt from *Ecolingua* in a formal language

From the facts in figures 2 and 3 the relations' instances in figure 4 are inferred. For example, $relation\_instance(eco\_data\_composition, tree, [dbh])$ represents an instance of the relation *eco_data_composition*, being *tree* an instance of the relation's domain class *ecological_data_entity*, and *[dbh]* a list containing an instance of the relation's range class *ecological_variable* (a relation can have more than one range class, in which case the list contains more than one element).

Figures 3 and 4 together depict the classes' instances and the relations that hold among them, expressing the following (which is compliant with our example's dataset properties): "*trees* are one of the *ecological data entities* sampled. The *data* about *trees is composed* by *ecological variables*. One of these variables is the *dbh of trees*, having *values* measured in *cm*. The number of *sampling points* is *360*, and each of them has been sampled *once* in *1990* and *once* in *1995*."

Now, recall that the parameter whose derivation we want to infer is the *average annual increment of dbh of trees*. The metadata formally described tell us that the variable *dbh* exists in the dataset, being an instance of the class *ecological_variable*. Moreover, the relations in the metadata between the classes (and its instances) provide the information needed for the derivation of the parameter.

$instance\_of(tree, ecological\_data\_entity).$
$instance\_of(dbh, ecological\_variable).$
$instance\_of(cm, unit\_of\_measure).$
$instance\_of(360, non\_negative\_integer).$
$instance\_of(times\_per\_year(1), samples\_per\_time).$
$instance\_of([year(1990), year(1995)], sampling\_times).$

Figure 3: Instances of *Ecolingua*'s classes in a formal language

$relation\_instance(eco\_data\_composition, tree, [dbh]).$
$relation\_instance(sampling\_points, dbh, [360]).$
$relation\_instance(sampling\_frequency, dbh,$
$[times\_per\_year(1), [year(1990), year(1995)]]).$

Figure 4: Instances of *Ecolingua*'s relations in a formal language

The first question to pose is how a human modeler would derive the *average annual increment of dbh of trees* from a dataset with the characteristics of our example dataset. One possible simple process would be:

1. Calculate the total increment of DBH of each individual tree during the 5 year period between the two measurements taken (1990 and 1995), which can be expressed as:

$$total(increment(dbh, Indiv)) = last\_value(dbh, Indiv) - first\_value(dbh, Indiv) \quad (1)$$

   Nested terms such as *total(increment(...* represent composition of functions, which are applied one after the other from the inner part of the term outwards. E.g. *increment* is calculated first, and then, *total* is calculated over *increment*.

2. Then, the modeler could approximate the annual increment of DBH of each tree by dividing the total increment calculated above by the number of years in the period between the two measurements:

$$annual(increment(dbh, Indiv)) = total(increment(dbh, Indiv))/5 \quad (2)$$

3. And finally, have the *average annual increment of dbh of trees* as the sum of the annual increment of DBH of all the 360 trees divided by the number of trees:

$$average(annual(increment(dbh, tree))) = sum(Indiv, 1, 360, annual(increment(dbh, Indiv)))/360 \quad (3)$$

One way of having a similar *metadata↔model* connection drawn automatically is to build a mechanism able to perform the process enumerated above. That is: given the metadata of an ecological dataset formally described as above, as well as the parameter to be derived *average annual increment of dbh of trees*, the mechanism *constructs* (or infers) equations 1, 2 and 3. Figure 5 shows a Definite Clause Grammar (DCG) that has this effect.

Let us explain the grammar's clauses together with some of the relations involved.

- *construct*'s first clause:
  The first clause *constructs* as terminal expression, with instantiated variables:

$$average(annual(increment(dbh, tree))) = sum(Indiv, 1, 360, annual(increment(dbh, Indiv)))/360$$

  (which is equation 3) and as non-terminal expression a recursive call which will *construct* the equation expressing how to derive, in turn, *annual(increment(dbh,Indiv))* (equation 2).
  The relation:

$$entity\_term(X, Eco\_entity)$$

  returns *tree* (in *Eco_entity*) which is the ecological data entity with which the parameter (in X) in question is concerned. What tells the mechanism that it is consistent to derive the *average annual increment of dbh* of the ecological data entity *tree* is the fact in the description of the dataset:

$$relation\_instance(eco\_data\_composition, tree, [dbh])$$

  The relation:

$$about\_entity\_term(X, X1, Eco\_var, Indiv)$$

  returns in $X1$ the term *annual(increment(dbh,Indiv))* which specifies the relevant calculation for the ecological data entity *tree* in the derivation of the *average* parameter, i.e., *annual increment of dbh of trees* needs to be calculated intermediately to allow the derivation of the *average*. By doing this, the relation points out the *ecological variable* under consideration, in this case *dbh*, and introduces the variable *Indiv* to characterise the individualised *dbh* measurements of trees.
  The relation:

$$cardinality(Eco\_var, Eco\_entity, N)$$

  returns how many values there are for the pair *(Eco_var,Eco_entity)*. From the description of the ecological dataset it is known that "the DBH of 360 trees has been measured", which gives a *cardinality* of *360* to *(dbh,tree)*. This knowledge is represented in the relations below, which are part of the description of the dataset:

$$relation\_instance(eco\_data\_composition, tree, [dbh])$$
$$relation\_instance(sampling\_points, dbh, [360])$$

- *construct*'s second clause:
  The second clause *constructs* as terminal expression, with instantiated variables:

$$annual(increment(dbh, Indiv)) = total(increment(dbh, Indiv))/5$$

$$
\begin{array}{l}
construct(Construction\_info, average(X)) \; {-}{-}{>} \\
\quad \{entity\_term(X, Eco\_entity) \;\wedge \\
\quad about\_entity\_term(X, X1, Eco\_var, Indiv) \;\wedge \\
\quad cardinality(Eco\_var, Eco\_entity, N)\}, \\
\quad [average(X) = sum(Indiv, 1, N, X1)/N], \\
\quad construct([cardinality(Eco\_var, Eco\_entity, N)|Construction\_info], X1). \\
\\
construct(Construction\_info, annual(X)) \; {-}{-}{>} \\
\quad \{cardinality(Eco\_var, Eco\_entity, N) \;\in\; Construction\_info \;\wedge \\
\quad total\_time\_span(year, Eco\_var, Eco\_Entity, Years)\}, \\
\quad [annual(X) = total(X)/Years], \\
\quad construct(Construction\_info, total(X)). \\
\\
construct(Construction\_info, total(increment(Eco\_var, Indiv))) \; {-}{-}{>} \\
\quad \{cardinality(Eco\_var, Eco\_entity, N) \;\in\; Construction\_info \;\wedge \\
\quad first\_time(Eco\_var, Eco\_entity, Tfirst) \;\wedge \\
\quad last\_time(Eco\_var, Eco\_entity, Tlast) \;\wedge \\
\quad for(Indiv, 1, N, value\_exists(Indiv, Eco\_var, Eco\_entity), Tfirst) \;\wedge \\
\quad for(Indiv, 1, N, value\_exists(Indiv, Eco\_var, Eco\_entity), Tlast)\}, \\
\quad [total(increment(Eco\_var, Indiv)) = last\_value(Eco\_var, Indiv) - first\_value(Eco\_var, Indiv)].
\end{array}
$$

Figure 5: DCG generating the derivation of the parameter *average annual increment of dbh of trees* from ecological metadata

and as non-terminal expression a recursive call which will *construct* the equation expressing how to calculate, in turn, *total(increment(dbh,Indiv))* (equation 1).

The relation:

$$total\_time\_span(year, Eco\_var, Eco\_entity, Years)$$

returns the total time span in years that has passed between the two measurements in 1990 and 1995. This knowledge is represented by the fact:

$$relation\_instance(sampling\_frequency, dbh, [times\_per\_year(1), [year(1990), year(1995)]])$$

- *construct*'s third clause:

  The third clause finalises the construction of the equations yielding the following terminal expression, with instantiated variables:

  $$total(increment(dbh, Indiv)) = last\_value(dbh, Indiv) - first\_value(dbh, Indiv)$$

  The relations:

  $$first\_time(Eco\_var, Eco\_entity, Tfirst),$$

  $$last\_time(Eco\_var, Eco\_entity, Tlast)$$

  give the time points (in this case the years 1990 and 1995) when the *dbh* measurements have been taken first and last respectively. The *relation_instance* for *sampling_frequency* in the dataset description is again used here.

  The relations:

  $$for(Indiv, 1, N, value\_exists(Indiv, Eco\_var, Eco\_entity), Tfirst),$$

  $$for(Indiv, 1, N, value\_exists(Indiv, Eco\_var, Eco\_entity), Tlast)$$

  check for the existence of a *dbh* value for each of the 360 *trees* for the first and last measurement time points respectively. Once more the *relation_instance* for *sampling_frequency* in the dataset description holds relevant knowledge as well as the relation:

  $$relation\_instance(sampling\_points, dbh, [N])$$

  that tells how many measurements have been taken in each time point.

Related Work

Logic-based approaches for ecological modeling have been proposed in (Robertson *et al.* 1991). Emphasis is placed on the use of domain knowledge to support modeling automation, making model assumptions explicit to enable more informed model analysis. Our work evolves from these ideas, adding to them by investigating how ecological metadata (which play a part in domain knowledge) can be conducive to model construction.

In (Uschold 1991) two key factors are pointed out as responsible for clogging the way of model construction in general: available modeling tools don't "cater for how users think about their problems (large conceptual distance)" and "the vast modelling search space". The core of the approach presented to tackle these problems (using ecological modeling as an example domain) consists of a language, Elklogic, based on typed lambda calculus, which is suitable for representing both domain and simulation modeling information. The representation requirements for Elklogic come from an ecological modeling "knowledge ontology", which is a point of reference to our work with *Ecolingua*. Elklogic also allows for attributes induction. It uses higher order functions to represent inferences such as 'if there is the attribute *weight* of a certain animal, it can be inferred that the attributes *average/total/maximum/minimum weight* apply to groups of the animal'. The outcomes of our research can contribute to the easing of the same problems addressed by (Uschold 1991). We believe that metadata (related to an ecological dataset at hand) plays an important role in the way modelers reason when modeling ecological systems, and that we can automatically populate the search space with modeling elements which can be justified in the metadata.

(Rickel & Porter 1997) reports on automated modeling of complex systems, having plant physiology as the evaluation domain. The system built, called TRIPEL, incorporates a new compositional modeling algorithm which, given a prediction (what if) question, the variables of the physical system, the influences among them, and other domain knowledge, outputs the simplest differential equation model that can adequately answer the prediction question. The domain knowledge used is found in a large multipurpose biology knowledge base. The novelty of our work, in comparison with the above and other leading composition modeling approaches, resides in having formal descriptions of datasets' meta-level properties as the core building blocks to the automated modeling task.

Discussion

The example presented, though exploratory, demonstrates automatic inference of a parameter derivation process based on ecological metadata and simple encoded knowledge about few ecological functions. By means of a simple logic-based formalism, we are able to reconstruct the inferences that a human modeler would possibly perform to derive the parameter. Moreover, this is all done relying on meta-level descriptions of the dataset, without taking into account *ecological data* in the conventional sense, i.e., actual values assigned to the variables involved (in this case, *dbh of trees*).

The enterprise of building an ontology for ecological metadata description through the Ontolingua Server, re-using off-the-shelf definitions from several other ontologies, has been time and effort consuming. We first started using the server as an experiment, believing that it would assist us in quickly constructing a well-engineered ontology. The frame-based (or object-oriented) representational language offered by the server's interface has been fairly expressive to capture the ecological concepts and relations we have considered in the design stage. However, for translating the designed ontology into Prolog, it was necessary to build additional tools to re-engineer the server's output. Future work will investigate to which extent a frame-based axiomatisation can actually support inferences of models from metadata, as well as how useful the definitions *Ecolingua* inherited from other ontologies are.

In summary, the class of problems addressed by this research is concerned with finding and automating connections between metadata and model substructure, through formal knowledge representations and inferences, having system dynamics ecological modeling as experimental domain. Results are expected not to be domain-dependent, and should be portable or adaptable across other physical systems modeling disciplines. The ultimate implementational goal is a system that semi-automates a range of system dynamics modeling tasks and is able to furnish the user with the underlying metadata support rationale. The system will infer prototypical models (or parts of them) to be interactively refined by human modelers.

Acknowledgments. To D. Robertson and R. Muetzelfeldt, my supervisors, for the guidance; to the anonymous reviewers, for valuable criticisms; and to the Brazilian Government - CAPES, grant BEX1498/96-7.

References

Biot, Y.; Higuchi, N.; Brilhante, V.; Freitas, J.; Ferraz, J.; Leal, N.; Ferreira, S.; and Desjardins, T. 1996. INFORM: the INpa FORest Model. Technical report, Project BIONTE, INPA - Brazil and ODA - UK, Manaus, Brazil.

Biot, Y. 1995. Data survey report - BIONTE Project. Technical report, INPA - Brazil and ODA - UK, Manaus, Brazil.

Farquhar, A.; Fikes, R.; and Rice, J. 1996. The ontolingua server: a tool for collaborative ontology construction. Technical Report KSL-96-26, Computer Science Department, Stanford University.

Gillman, M., and Hails, R. 1997. *Introduction to Ecological Modelling: Putting Practice into Theory.* Blackwell Science Ltd.

Gruber, T. R. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2):199–220.

Rickel, J., and Porter, B. 1997. Automated modeling of complex systems to answer prediction questions. *Artificial Intelligence Journal* 93(1-2):201–260.

Robertson, D.; Bundy, A.; Muetzelfeldt, R.; Haggith, M.; and Uschold, M. 1991. *Eco-Logic: logic-based approaches to ecological modelling.* The MIT Press.

Uschold, M., and Gruninger, M. 1996. Ontologies: principles, methods and applications. *The Knowledge Engineering Review* 11(2):93–136.

Uschold, M. 1991. The use of domain information for comprehension and construction of simulation models. Technical Report DAI Research Paper 534, Department of Artificial Intelligence, University of Edinburgh.