Disaggregation in Compositional Modelling of Ecological Systems via Dynamic Constraint Satisfaction

Jeroen Keppens & Qiang Shen
Division of Informatics, University of Edinburgh
Email: {jeroen,qiangs}@dai.ed.ac.uk

Abstract

This paper is concerned with the use of compositional modelling for effectively building and (re-)using repositories of ecological models, with a focus on the important issue of model granularity. The techniques presented herein differ from existing work in that a meta-modelling approach to grain choices is taken by transforming an emerging model's level of detail through disaggregation. The result is a means of choosing a model's level of detail that is sufficiently flexible for eco-modelling and that allows grain choices to be described in terms of scenario-level concepts. Model disaggregations are implemented as model transformations. Thus, different disaggregations of a model can be composed themselves. As such, a small number of disaggregations may encompass a much larger selection of models at different levels of detail. The paper shows that due to the use of dynamic constraint satisfaction techniques, grain choices are not necessarily made from the outset, but can be postponed until after an initial model is constructed.

Introduction

Compositional modelling techniques (Falkenhainer, B. & Forbus, K.D. 1991; Keppens, J. & Shen, Q. 2001) offer tools for explicit knowledge representation and to facilitate model based reasoning. Although conventionally designed for physical systems, compositional modellers have been recently applied to ecological modelling (Struss 1998). This paper presents work following this line of development. In particular, it aims at building a tool for establishing model repositories and exploring modelling alternatives of eco-systems, based on the existing work reported in (Keppens, J. & Shen, Q. 2000).

By means of dynamic constraint satisfaction, the existing approach allows ecological models of different topology, scope and/or involving alternative types of relation to be constructed with respect to a given scenario. However, the important issue of model granularity has not been resolved. The granularity of a model is the level of detail at which it describes a system. Some compositional modellers, aimed at physical systems, tackle granularity by defining a component hierarchy from which an appropriate level of detail can be selected. Alternative approaches allow variables and influences of different grain sizes to be defined and the knowledge base prescribes how these relate to one another. The former approach is more comprehensible in that it can express granularity in terms of the inherent concepts of the scenario under consideration. Yet, it makes the assumption that the concepts being modelled can be organised into a component hierarchy, which does not generally hold for ecological systems. The latter approach is more generic because it can be applied to systems that are conventionally modelled by means of processes as well as components. Unfortunately, because the grain choices are embedded in the variables and their relations (typically represented as equations) they cannot be named in terms of scenario-level concepts.

It is important for an ecological model repository to satisfy both the criteria of comprehensibility and generality. To this end, this paper incorporates grain choices through *disaggregation*. A disaggregation is a model transformation that involves replacing variables and/or equations in a model by sets of variables and/or equations, each representing a partition of parts of the concepts/processes described by the original variables/equations. This entails the change of model granularity. The work allows for a variety of disaggregations of processes and components and for them to be related to the scenario level concepts. Furthermore, it enables disaggregations to be automatically composed with one another, such that the knowledge base can be specified with a minimal amount of redundancy.

The work presented herein incorporates disaggregation into compositional modelling, again, by means of dynamic constraint satisfaction. Through this, disaggregation can be dynamically made after an initial model has been constructed, by repairing and extending the previous solution. This is particularly desirable in eco-modelling as a model repository is typically bound to be used in a relatively interactive fashion.

Basic concepts

Definition 1 (*Model*). A *model* μ is a tuple $\langle P, R \rangle$ where P is a set of object constants and R is a set of *relations* over these objects. The object constants in P are called *participants*, since they refer to objects that participate in the model.

The objects referred to by the participants and relations may be interpreted as real-world objects and interrelationships, such as a population and predation, or conceptual objects and interrelationships, such as a variable representing population size and an equation modelling predation as a function of predator population size. A representation (which is itself a model) of only real-world objects and interrelationships is called a *scenario*. The purpose of compositional modelling is to infer a mathematical model μ_M from a given scenario μ_S by committing a consistent set of assumptions A. The knowledge that us used to deduce $\mu_S, A \vdash \mu_M$ consists of model fragments.

Definition 2 (*Model fragment*). A *model fragment* is a tuple $\langle P^s, P^t, A, \Phi^s, \Phi^t \rangle$ where the source-participants $P^s = \{\ldots, P^s_i, \ldots\}$ and the target-participants $P^t = \{\ldots, P^t_i, \ldots\}$ are sets of participant classes[1], the structural conditions $\Phi^s = \{\ldots, \phi^s_i, \ldots\}$ and assumptions $A = \{\ldots, a_i, \ldots\}$ are sets of relations[2] defined over $P^s_q \times \ldots \times P^s_r$ and the set of postconditions $\Phi^t = \{\ldots, \phi^t_i \ldots\}$ is a set of relations defined over $P^s_q \times \ldots \times P^s_r \times P^t_v \times P^t_w$, such that $\forall p^s_1 \in P^s_1, \ldots, p^s_m \in P^s_m, \exists p^t_1 \in P^t_1, \ldots, p^t_n \in P^t, (\wedge_i \phi^s_i(\overrightarrow{p_i}) \rightarrow (\wedge_j a_j(\overrightarrow{p_j}) \rightarrow \wedge_k \phi^t_k(\overrightarrow{p_k})))$.

As described in (Keppens, J. & Shen, Q. 2000), the participants and relations that can be derived from a given scenario μ_S and a knowledge base Γ of model fragments is formalised in a hypergraph, called the *model space*. Such a hypergraph has a root node for each of the participants and relations of the scenario, a root node for each instantiated assumption and a derived node for each of all the other instantiated participants, relations and model fragments. The hyperarcs between these nodes describe how new participants and relations can be derived from existing ones. More specifically, for each instantiation of a model fragment, a hyperarc connects the nodes that instantiate the source-participants, structural conditions and assumptions with the node representing the model fragment instantiation. The latter node is in turn connected by arcs to nodes that instantiate the target-participants and postconditions.

To construct a scenario model efficiently, this model space is translated into a dynamic constraint satisfaction problem (Mittal, S. & Falkenhainer, B. 1990) or DCSP. A DCSP consists of:

- A set of *attributes* $\mathbf{X} = \{x_1, \ldots, x_n\}$. An attribute x_i is either active (denoted as active(x_i)) or inactive ($\neg$active(x_i))
- A *domain* for each attribute: $\mathbf{D} = \{D_1, \ldots, D_n\}$ where $D_i = \{v_{i1}, \ldots, v_{in_i}\}$ is a set of possible values of that attribute. Each active attribute x_i is assigned a value d_{ik_i} from its domain, denoted as $x_i : d_{ik_i}$.
- A set of *activity constraints* $\mathbf{C^a} = \{C^a_1, \ldots, C^a_{m^a}\}$ where C^a_l is an implication $x_i : d_{il_i} \wedge \ldots \wedge x_j : d_{jl_j} \rightarrow$ active(x_k).
- A set of *compatibility constraints* $\mathbf{C^c} = \{C^c_1, \ldots, C^c_{m^c}\}$ where C^c_k is a relation $D_i \times \ldots \times D_j \rightarrow \{\top, \bot\}$. If $C^c_k(d_{ik_i}, \ldots, d_{jk_j}) = \top$, then any set of assignments S, such that $\{x_i : d_{ik_i} \ldots x_j : d_{jk_j}\} \subseteq S$, is said to *satisfy* C^c_k.

Table 1 summarises how a model space is translated into a DCSP (see (Keppens, J. & Shen, Q. 2000) for a more detailed description). In essence, assumptions that represent alternatives of the same phenomenon or model are grouped into *assumption classes*. These assumption classes correspond to DCSP attributes whereas the individual assumptions correspond to domain values. From the structure of the model space, the activity constraints are derived. Certain relations (i.e. instantiated postconditions in the model space) cannot be composed into the same model, e.g. because they assign different equations to the same variable. Furthermore, certain properties can be defined about the variables and relations in the model space, e.g. the conditions that make a variable endogenous. Both of these features are translated into compatibility constraints.

A *solution* to a DCSP is a set of assignments such that all the activity constraints and all the compatibility constraints whose attributes are active are satisfied. This DCSP solution implies a composed model that follows from the model space under the assumptions that are equivalent to the attribute-value assignments. Figure 1 shows a typical system dynamics model that can be constructed by this early version of the DCSP-based compositional modeller. It represents an ecological system in which a predator population feeds on a prey population, using the logistic population growth model (Verhulst 1838) and a predation model based on Hollings disc equation (Holling 1959).

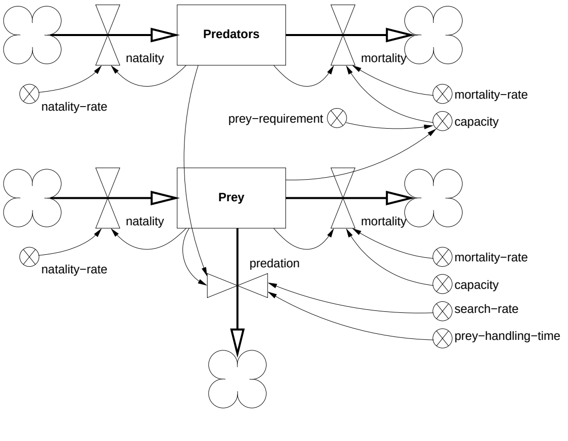

Figure 1: Sample system dynamics model of an ecological system

[1] A participant class is a well-formed formula that defines a set of participants. In this paper, without causing confusion, participant classes are referred to as actual sets of participants.

[2] In what follows, instances of relations $r(p_r, \ldots, p_q)$ are denoted as $r(\overrightarrow{p})$ and domains of relations $P_r \times \ldots \times P_q$ are denoted as $\overrightarrow{P}$.

CM or model space concept	DCSP concept
Assumption class	Attribute
Assumption	Domain value
Set of assumptions that support a model fragment instance	Activity constraints
Inconsistent relations	Compatibility constraints
Properties	Compatibility constraints

Table 1: Translating a model space into a DCSP

Granularity of ecological models

The compositional modeller described above is suitable to compose models of different phenomena and of model types given a fixed scenario and level of detail. However, the issue of *granularity* has not been sufficiently addressed. Depending on the modelling task at hand, ecologists may require a model to represent certain features in more detail than previously modelled. For example, the predator population may need to be divided into different age categories because the subpopulations representing different feeding requirements and reproduction patterns may be required. Or, the prey population could consist of different subspecies with different characteristics of, say, reproduction rate and running speed and hence the ability of avoiding being hunted up. Also, it may become necessary to model explicitly the spatial distribution of the vegetation because of an interest in migration is established at the later stage of modelling.

These phenomena involving more detailed information or smaller granularity could still be modelled by means of the representations discussed in the previous section. However, composition of model fragments, describing ecological processes at different granularities, is not straightforward. When instead of a single participant, e.g. that representing the total population size, a number of participants of smaller grain are needed, e.g. those representing population sizes of different age classes, a significant number of phenomena such as growth, predation and competition are affected. New model fragments must be defined for all of these. To address this type of modelling problem, the approach taken herein is to capture the intuitive model transformation mechanism, which are typically used by ecologists, and to represent it explicitly as disaggregations of the most basic case.

Concept of model disaggregation

The purpose of disaggregating an existing model into a more detailed one is to provide a concise way of expressing different phenomena, i.e. the participants and relations that describe them, with varying granularities. The grain size or granularity refers to the level of detail at which populations, instead of individual components, and the phenomena that are relevant to them are described. For this reason, increasing the level of detail is hereafter referred to as disaggregation into classes. The classes are subpopulations of individuals that are similar with regard to certain attributes, such as age, sex, species or social status, and that are being considered as the result of disaggregation.

In general, disaggregating a model into n classes involves one or more of the following:

1. Replacing a certain participant by a set of n participants.
2. Mapping the relations of the disaggregated participants onto disaggregate relations. That is, each relation $r(p_1^d, \ldots, p_m^d, p_1, \ldots, p_n)$, where $p_1^d, \ldots, p_m^d$ are the disaggregated participants, is replaced by n relations $r(p_{i1}^d, \ldots, p_{im}^d, p_1, \ldots, p_n)$, $i = 1, \ldots, n$.
3. Adding additional participants and relations. These are typically used to describe migration between the resulting disaggregate classes.

Definition 3 (*Disaggregate model*). A model $\langle P^d \cup P^c \cup P', R^d \cup R \cup R' \rangle$ is said to be a *disaggregate* one of another $\langle P^a \cup P^c, R^a \cup R \cup R'' \rangle$ if there exits a surjection $\sigma : P^d \rightarrow P^a$ and a injection $\rho : \mathrm{dom}(\rho) \rightarrow R^a$ (with $\mathrm{dom}(\rho) \subset R^d$), such that $\forall p^a \in P^a, \exists \{p_1^d, \ldots, p_n^d\} \subset P^d, (\sigma(p_1^d) = p^a \wedge \ldots \wedge \sigma(p_n^d) = p^a)$ and $\forall r(p_q^d, \ldots, p_r^d, p_v^c, \ldots, p_w^c) \in R^d - \mathrm{dom}(\rho), r(\sigma(p_q^d), \ldots, \sigma(p_r^d), p_v^c, \ldots, p_w^c) \in R^a$. The relations in R^a are said to be *disaggregated* into the disaggregate relations of R^d (as described in item 2 above).

For example, the simple model of logistic population growth (Verhulst 1838) can be summarised by the following differential equations:

$$\frac{d}{dt}N = C^+(B), \frac{d}{dt}N = C^-(D) \tag{1}$$

$$B = r \times N, D = d \times N \times \frac{T}{K}, T = C^+(N) \tag{2}$$

where N is the population size, B is the number of births within a given time interval, D is the number of deaths within the same time interval, r is the reproduction rate, d is the death rate, T is the total relevant population and K is the population capacity. A disaggregate model of this logistic growth model with respect to, say, n age classes may be:

$$\frac{d}{dt}N_0 = C^+(B_i), \frac{d}{dt}N_i = C^-(D_i) \tag{3}$$

$$B_i = r_i \times N_i, D_i = d_i \times N_i \times \frac{T}{K}, T = C^+(N_i) \tag{4}$$

$$\frac{d}{dt}N_i = C^-(M_i), M_{i+1} = \mathrm{delay}(N_i, t_i) \tag{5}$$

In the aggregate model, $P^a = \{N, D, r, d\}$, $P^c = \{B, T, K\}$ and equations (1) and (2) are relations in R^a. In accordance with definition 3, the participants of P^a are mapped to $P^d = \{N_0, \ldots, N_n, D_0, \ldots, D_n, r_0, \ldots, r_n, d_0, \ldots, d_n\}$, those in P^c are retained and the equations in R^a are mapped onto (3) and (4) respectively, which are the relations in R^d. Migration is described by the set of participants $P' = \{M_1, \ldots, M_n\}$ and by the set of relations containing the equations expressed by (5).

Representing and inferring disaggregation

The purpose of the compositional modeller is not to disaggregate individual models but to produce correctly disaggregated models. As argued earlier, it is not appropriate to represent such knowledge directly by means of model fragments because this would result in much duplication of knowledge which makes the knowledge base harder to construct and maintain. This would also cause overly large model spaces to be constructed that complicate the search for a solution for problems that do not require consideration of all possible disaggregations. More importantly, in eco-modelling, what model or part of a model is needed to be disaggregated may not be known when the modelling task starts. It is usually required to disaggregate during or after the modelling process has been initiated as the ecologists may feel there is a need to examine further details of certain part(s) of the phenomenon being modelled. To facilitate the required dynamicity in knowledge representation, the concepts of disaggregation mapping and disaggregation fragments are introduced.

Definition 4 (*Disaggregation mapping*). A *disaggregation mapping* M is a tuple $\langle N, \delta_P, \delta_R \rangle$ where

- $N = N_1 \times \ldots \times N_l$, with N_i being a set of integers that represent names of classes,
- δ_P is a bijection $N_1 \times \ldots \times N_l \times \text{dom}(\delta_P) \to \text{range}(\delta_P)$ $(\text{dom}(\delta_P), \text{range}(\delta_P) \subset \mathcal{P}$ and $\mathcal{P}$ is the set of all participants),
- δ_R is a bijection $N_1 \times \ldots \times N_l \times \text{dom}(\delta_R) \to \text{range}(\delta_R)$ $(\text{dom}(\delta_R), \text{range}(\delta_R) \subset \mathcal{R}$ and $\mathcal{R}$ is the set of all relations),

such that $\forall p \in \text{dom}(\delta_P), (\forall i \neq j, \delta_P(\ldots, i, \ldots, p) \neq \delta_P(\ldots, j, \ldots, p))$,

In terms of definition 3, the participant mapping δ_P states which participant classes belong to P^a and how they are mapped onto P^d of the disaggregate model. For what follows, a transformation called the generalised participant mapping $\delta'_P : \mathcal{P} \to \mathcal{P}$ is defined such that for all $p \in \text{dom}(\delta_P), \delta'_P(\ldots, n_i, \ldots, p) = \delta_P(\ldots, n_i, \ldots, p)$ and all $p \in \mathcal{P} - \text{dom}(\delta_P), \delta'_P(\ldots, n_i, \ldots, p) = p$. The bijection δ_R describes how the relations of R^a in definition 3 that do not disaggregate according to the defined classes map onto R^d.

For the above example of disaggregating a population into age classes, a suitable disaggregation mapping is:

$$\delta_P(\overrightarrow{i}, N) = N_{\overrightarrow{i}}, \delta_P(\overrightarrow{i}, D) = D_{\overrightarrow{i}}, \tag{6}$$

$$\forall p, \text{parameter}(p) \to \delta_P(\overrightarrow{i}, p) = p_{\overrightarrow{i}}, \tag{7}$$

$$\delta_R(\overrightarrow{i}, (\frac{d}{dt}N = C^+(B))) = (\frac{d}{dt}N_0 = C^+(B)) \tag{8}$$

This disaggregation mapping states that N,B and D and the parameters b and d are mapped into n age classes. It furthermore specifies a specific mapping for $\frac{d}{dt}N = C^+(B)$ in (1) to $\frac{d}{dt}N_0 = C^+(B)$ in (3).

Definition 5 (*Disaggregation fragment*). A disaggregation fragment is a tuple $\langle P^s, P^t, A, \Phi^s, \Phi^t, M \rangle$ where

- $P^s = \{P^s_1, \ldots, P^s_m\}$ is a set of participant classes, called *source-participants*,
- $P^t = \{P^t_1, \ldots, P^t_n\}$ is a set of participant classes, called *target-participants*,
- A is the set of assumptions that a disaggregation fragment depends on. All assumption specifications in A refer to assumption instances other than those referred to in the (normal) model fragments in the knowledge base (see below).
- Φ^s is a set of relations defined over $P^s_q \times \ldots \times P^s_r$, called *structural conditions*,
- Φ^t is a set of relations defined over $P^s_q \times \ldots \times P^s_r \times P^t_v \times P^t_w$, called *target participants*, and
- M is a disaggregation mapping

such that $\forall \langle P^a, R^a \rangle \in \{\mu_M \mid \mu_S, A \cup \{\neg\text{disaggregation}(P(M))\} \vdash \mu_M\}$, $(\mu_S, A \cup \{\text{disaggregation}(P(M))\} \vdash \langle P^d, R^d \rangle)$, where $P^d = \{p^d \mid p^d = \beta(n_1, \ldots, n_l, p^a), p^a \in P^a\} \cup P^t$ and $R^d = \{r(\delta'_P(n_1, \ldots, n_l, p^a_q), \ldots, \delta'_P(n_1, \ldots, n_l, p^a_r)) \mid \wedge(r(p^a_q, \ldots, p^a_r) \in R^a) \wedge (r(p^a_q, \ldots, p^a_r) \notin \text{dom}(\delta_R))\} \cup \{\delta_R(r(p^a_q, \ldots, p^a_r)) \mid (r(p^a_q, \ldots, p^a_r) \in \text{dom}(\delta_R))\} \cup \Phi^t$

```
(defDisaggregationFragment population-age-classes
  :source-participants ((?p :type population)
                        (?pn :type stock :unit population)
                        (?pb :type flow :unit population)
                        (?pd :type flow :unit population))
  :structural-conditions ((stock ?p ?pl)
                          (flow ?pb source ?pn)
                          (flow ?pd ?pn sink))
  :meta-participants ((?n :type integer))
  :assumptions ((disaggregation ?p age-classes ?n))
  :mapping-types (((age-classes ?t) :type (array (0 ?n) ?t)))
  :target-participants
   ((?pn* :type (age-classes stock) :mapped-from ?pn)
    (?pd* :type (age-classes flow) :mapped-from ?pd)
    (? :type (age-classes parameter) :mapped-from (? :type parameter))
    (?pm :type (array (1 ?n) flow))
    (?ts :type (array (0 (1- ?n)) variable)))
  :postconditions
   (((flow ?pb source (?pn 0)) :mapped-from (flow ?pb source ?pn))
    (for (?i 1 ?n) (flow (?pm (1- ?i)) (?pn (1- ?i)) (?pn ?i)))
    (for (?i 1 ?n) (== (?pm i) (delay (?pn (1- ?i)) (?ts ?i))))))
```

Figure 2: Disaggregation fragment for population age classes

Figure 2 shows the disaggregation fragment for the ongoing example of splitting up a population into n age classes. It states that this disaggregation is applicable to a population `?p` for which a stock `?pn` and two flows `?pb` and `?pd` are known, such that `?pb` is the flow into `?pn` and that `?pd` is the flow out of `?pn`. Both the `predator` and the `prey` populations in figure 1 can be modelled in this way. The following changes are needed to the part of the model space that can be derived from the model fragments which have been used to produce the source-participants and structural conditions of the disaggregation fragment:

- The labels $\mathcal{L}(mf_i)$ of the instances of the model fragments mf_i that derived the instances of the source-participants and structural conditions is replaced by $\mathcal{L}(mf) \wedge (\neg\text{disaggregation}(\texttt{?p}, \text{age-classes}, \_))$. The participants and relations derived below will depend on the set of assumptions $\mathcal{L}(mf) \wedge (\text{disaggregation}(\texttt{?p}, \text{age-classes}, ?n))$
- Any participants, relations or model fragments that are derived from the original model fragments mf_i are transformed according to the disaggregation mapping introduced earlier. The `mapping-types` item defines the disaggregation of a participant type `?t` into an array of `?n+1` items of type ?t. As given in the example, the population size (`?pn`), the deaths (`?pb`) and all parameters are mapped onto such an array.
- The flow of births $\frac{d}{dt}N = C^{+}(B)$, represented by `(flow ?pb source ?pn)` in the knowledge base, is mapped to $\frac{d}{dt}N_0 = C^{+}(B)$, represented by `(flow ?pb source (?pn 0))` in the disaggregation fragment.
- An array of `?n` flows `?pm` are added to represent migration from one age class to the next. Equation (5) of the disaggregate model is described by the last two postconditions of figure 2.

Intuitively it is clear how an expert human modeller may derive this disaggregate model from given disaggregation fragments. The compositional modeller developed in this work formalises the intuitions as explained below.

Inferring disaggregations

Similar to the normal model fragments, disaggregation fragments are applicable with respect to a set of instances of source-participants and structural conditions. However, the application of a disaggregation fragment requires copying and transforming all of the participants and relations that depend upon the model fragments which have instantiated the source-participants and the structural conditions.

The algorithm ApplyDF(d, M) describes the procedure of extending and transforming the model space for an application of a disaggregation fragment d, which is applicable with respect to the set of model fragment instances M. In less abstract terms, the example given in figure 3 shows the result of applying ApplyDF(d, M) to the disaggregation fragment of figure 2, with respect to an instance of a model fragment for population growth. ApplyDF(d, M) essentially copies the subtree of consequents of M, disaggregates all participants and relations according to the bijections δ_P and δ_R of the disaggregation mapping respectively, and adds the new participants and relations for migration.

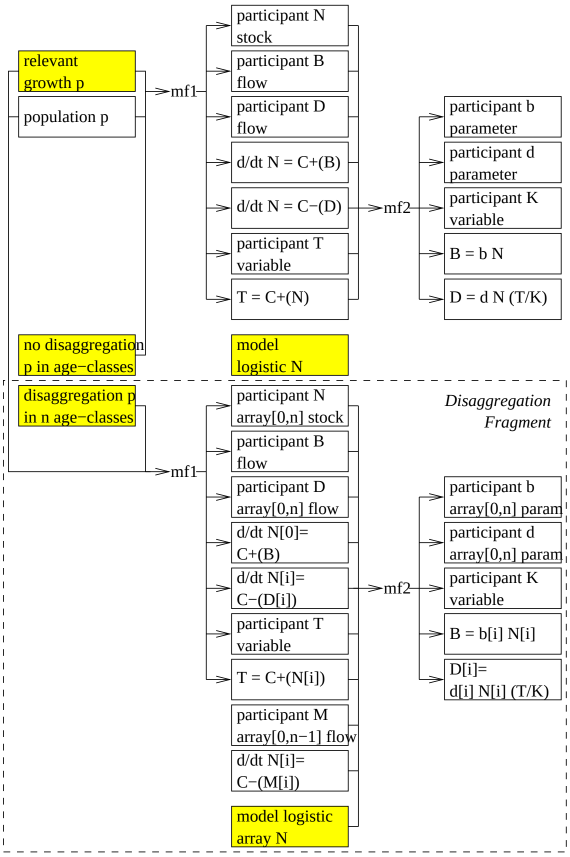

Figure 3: Model space expansion via disaggregation fragment

In particular, the upper half of figure 3 shows the model space generated using the model fragments for the population growth phenomenon (mf1) and the one for the logistic growth model (mf2) with respect to a single population. Under the set of assumptions “relevance of growth of p”, “logistic growth model for N” and “no disaggregation of p in age classes”, the model specified in equations (1) and (2) follows. The bottom half of the figure is the disaggregated version of the model space added due to the application of the aforementioned disaggregation fragment. By committing the assumptions “relevance of growth of p”, “logistic growth model for N” and “disaggregation of p into n age classes”, the model specified in equations (3), (4) and (5) follows.

Theorem 6. If a model M can be derived from a model space Δ, M can also be derived from Δ' representing Δ extended by the application of a disaggregation fragment *df*
Proof not included in this version due to space limitations. ■

Theorem 7. Given that (i) Δ is a model space, (ii) Δ'

is the model space resulting from extending Δ by the application of a disaggregation fragment *df* to a set of model fragment nodes Mf, (iii) A is a set of assumptions such that $A, \Delta \nvdash \bot$, and (iv) M_A is a model such that $A, \Delta \vdash M_A$, then the model M_D such that $A \cup A_{df}, \Delta' \vdash M_D$ is a disaggregate model of M_A
Proof not included in this version due to space limitations. ■

```
ApplyDF($d, M$);
begin
   $Q := M;\ S := \{\}$;
   repeat
   $m :=$ dequeue($M$);
   $m' :=$ new model fragment instance node;
   if$m \in M$
   then
   begin
      justify($m', \mathcal{L}(m) \wedge d(A)$);
      justification($m$) $:= \mathcal{L}(m) \wedge \neg d(A)$
   end
   else justify($m'$, substitute($A(m), S$));
   for $p \in$ consequents($m$), participant($p$) do
         if $p \in$ dom($\delta_P$)then
         begin
            $S := S \cup (p, \delta_P(p))$;
            $n :=$ node($\delta_P(p)$);
         end;
         else$n :=$ node($p$);
         justify($n, m'$);
         $\forall m''$, (successor($m'', p$) $\rightarrow$ enqueue($m'', Q$))
   end for;
   for $\phi(\overrightarrow{p}) \in$ consequents($m$), relation($\phi(\overrightarrow{p})$) do
         if $\phi(\overrightarrow{p}) \in$ dom($\delta_R$)
         then $n :=$ node($\delta_R(\phi(\overrightarrow{p}))$);
         else $n :=$ node(disaggregate($\phi(\overrightarrow{p}), S$));
         justify($n, m'$);
         $\forall m''$, (successor($m'', p$) $\rightarrow$ enqueue($m'', Q$));
   end for;
   if $m \in M$then
   begin
      $\forall p \in P^t(d)$, ($n :=$ node($p$), justify($n, m'$));
      $\forall \phi(\overrightarrow{p}) \in \Phi^t(d)$, ($n :=$ node($\phi(\overrightarrow{p})$), justify($n, m'$));
   end;
   until$M = \{\}$;
end
```

Algorithm 1: Extending the model space

Combining disaggregations

Up to now, only individual disaggregations have been discussed. There are, however, many scenarios where it may be necessary to apply different disaggregations to the same participant. For example, in addition to disaggregating a population into age classes, a population could be disaggregated according to sex, physical location or subspecies. The effects of these disaggregations must therefore be combined.

Definition 8 (*Combined application of two disaggregation fragments*). The combined application of two disaggregation fragments to a model fragment instance, denoted ApplyDF($d_2 \circ d_1$,M), involves (i) applying d_2 to the set of model fragments generated by applying d_1 to M and (ii) applying the disaggregation mapping of d_1 to the instances of the target-participants and postconditions introduced into the model space by applying d_2.

Consider, for example, disaggregation into q populations of subspecies. This requires a disaggregation mapping that disaggregates all participants other than T (total population) and K (maximal sustainable population). The application this disaggregation mapping to equations (1) and (2) results in (with $j = 0, \ldots, q$):

$$\frac{d}{dt}N_j = C^+(B_j), \frac{d}{dt}N_j = C^-(D_j) \tag{9}$$

$$B_j = r_j \times N_j, D_j = d_j \times N_j \times \frac{T}{K}, T = C^+(N_j) \tag{10}$$

Now consider combining this with the disaggregation into n age-classes discussed earlier. Following definition 8, applying age-class disaggregation to (9) and (10) involves (i) applying the age-class disaggregation fragment and (ii) applying the subspecies disaggregation fragment to the target-participants and postconditions introduced by age-class disaggregation. More specifically, step (i) results in:

$$\frac{d}{dt}N_{0j} = C^+(B_{ij}), \frac{d}{dt}N_{ij} = C^-(D_{ij}) \tag{11}$$

$$B_{ij} = r_{ij} \times N_{ij}, D_{ij} = d_{ij} \times N_{ij} \times \frac{T}{K}, T = C^+(N_{ij}) \tag{12}$$

$$\frac{d}{dt}N_i - 1 = C^-(M_i), M_i = \text{delay}(N_i, t_i) \tag{13}$$

with $i = 0, \ldots, n$ and $j = 0, \ldots, q$ in (11) and (12) and with $i = 0, \ldots, n$ in (13). Step (ii) applies to (13) and results in:

$$\frac{d}{dt}N_{i-1,j} = C^-(M_{ij}), M_{ij} = \text{delay}(N_{ij}, t_{ij}) \tag{14}$$

Theorem 9. ApplyDF($d_2 \circ d_1$,M)=ApplyDF($d_1 \circ d_2$,M)
Proof not included in this version due to space limitations. ■

Due to theorem 9, the combined application of disaggregation fragments is commutative. Therefore, the significance of using disaggregation fragments to assist combining disaggregations lies in the fact that, if the disaggregation fragments are sufficiently general, combinations amongst them are implemented automatically. If, however, only model fragments were used to specify disaggregations a different set of model fragments would be necessary for each combination of disaggregations. This is because each combination implies a different, whilst similar, set of participants and relations. As disaggregation fragments can be composed, only one is needed for each type of disaggregation instead of one per combination of disaggregations.

Dynamic Constraint Satisfaction

The extended model space produced by applying disaggregation fragments to an existing model space can be translated into an CSP in the same way as discussed in (Keppens, J. & Shen, Q. 2000). This CSP contains activity constraints within it that restrict the assignments under which attributes, and the constraints containing these attributes apply (Mittal, S. & Falkenhainer, B. 1990). Such a CSP is therefore called an internally dynamic CSP (iDCSP). However, the model space may grow exponentially when all disaggregation fragments are automatically applied and combined. Thus, it is necessary to disaggregate only when the user presents such requests. This setting poses a CSP that allows the attributes, domains and constraints to change due to requirements imposed externally over the initially specified CSP. Such kind of dynamic CSP will be referred to as an externally dynamic CSP (eDCSP). As with any DCSPs, solving an eDCSP requires solution techniques that can repair an existing solution (Dechter, R. & Dechter, A. 1988; Miguel, I. & Shen, Q. 2000). This section describes how finding a solution to a CSP which is internally and externally dynamic leads to the construction of a disaggregate model.

Note that expanding a model space with respect to the application of a disaggregation fragment to a set of model fragment instances may result in the following changes to the model space:

1. The new assumptions in the transformed model subspace are grouped into new assumption classes. These may include the assumption class as containing $\neg$disaggregation(subject,_) and disaggregation(subject,$\overrightarrow{n}$), where $\overrightarrow{n}$ covers all possible numbers of disaggregation classes and the subject refers to the (existing scenario-level) participant or relation that is being disaggregated (e.g. predator-population) and the kind of disaggregation applied to it (e.g. disaggregation into age-classes).
2. The justifications of certain existing nodes are extended with an additional (negated) assumption.
3. The affected subgraph in the model space is transformed with respect to the disaggregation fragment and forms a new subgraph containing additional participants, relations and model fragment instances and uses the new assumptions and assumption classes.
4. The new subgraph may contain new inconsistencies within itself or with respect to an existing part of the original model space. Note that inconsistencies arise from assignments to the same exogenous variable that cannot be composed.

By following the guidelines summarised in table 1, these extensions require alterations to the iDCSP respectively as follows:

1. For each new assumption class, a new attribute and domain is added to the iDCSP. The domain contains a value for each assumption in the assumption class. The domain of disaggregation assumption class contains a value for the negated assumption and one value for each possible $\overrightarrow{n}$. For example, the assumptions on disaggregating a population into age-classes can be represented by the following domain: $\{\neg\text{disaggregation}(\texttt{?p}, \text{age-classes}, \_), 0, \ldots, n\}$.
2. Adding an assumption to a justification potentially affects all nodes that can be derived from that justification. The new label, i.e. the union of the assumption sets supporting a justification, is propagated as if a new justification is added to a node the model space. This may affect the labels of the participants, relations and model fragment instances, and hence, cause the antecedent of activity constraints to change. The propagation may also extend the label of the node representing inconsistency, which is translated to a new compatibility constraint per inconsistency. The implementation of iDCSP reads the activity and compatibility constraints directly from the model space, and hence, the label propagation takes effect immediately.
3. The subgraph that extends the model space because of the application of a disaggregation fragment is translated into an iDCSP in exactly the same way as explained in (Keppens, J. & Shen, Q. 2000).
4. As each assignment relation in the newly transformed subgraph is added, it is compared with already stored assignment relations of the same exogenous variable and any resulting inconsistency is reported to the model space.

Algorithms for finding solutions to the internally and externally dynamic constraint satisfaction problems already exist. For instance, the work reported in (Miguel, I. & Shen, Q. 2000) tackles the dynamicity by means of local repair techniques (Verfaillie, G. & Schiex, T. 1994). The employment of such an algorithm enables a limited form of model construction and selection. This is because they typically handle prioritised constraints and assignment preferences by the use of maximum and minimum operators. However, the specific requirement for combining processes and modelling choices at multiple levels presented by eco-modelling often demands more sophisticated combinations of preferences. For example, the worth of models of population growth does not only depend on their intrinsic usefulness, but also on the modelling choices (e.g. predation and competition phenomena, and models thereof) which they can be combined with. As min-max operators have the limitation of missing detailed information conveyed by those constraints that have a preference degree other than the smallest or largest value. To entail better forms of model selection, work on building solution techniques that can handle both internally and externally dynamic constraint satisfaction problems, with assignment preferences allowed to be combined in a more detailed manner, is currently ongoing.

Conclusions

This paper has presented ongoing work in the development of compositional modelling techniques for building repositories of modelling alternatives of ecological systems. As such, this work is complementary to the important existing work on compositional modelling of ecological systems, presented in (Heller, U. & Struss, P. 1998), which performs diagnosis and therapy recognition of ecological systems. It achieves this by considering multiple plausible configurations of an ecosystem through alternative sets of closed-world assumptions, but without considering multiple modelling alternatives with respect to any individual configuration.

A method to construct ecological models at different levels of detail has been proposed. This method differs from other techniques in that it takes a meta-modelling approach to granularity by changing an emerging model's level of detail through disaggregation. It is sufficiently flexible to describe grain choices and represent them in terms of scenario-level objects and relations. The knowledge representation scheme proposed to enable disaggregation does not contain an alternative set of model fragments for each level of detail but describes the underlying transformation. Not only is this a more concise form of representation, it automatically entails the composition of disaggregations also.

The possibility of combining disaggregation fragments could potentially lead to an explosion of the model space, however. In order to prevent this, disaggregation may need to be postponed until a basic model has been constructed and an expert user asks for a more appropriate disaggregation. It is indeed quite common for the ecologist-user to make requests for more detailed scenario representations interactively and do so only after an initial model has been constructed (Robertson, D. *et al.* 1991). This paper has explained how such model composition task can be treated as externally dynamic constraint satisfaction problems for which existing solution techniques can be applied to select and construct disaggregate models.

Nevertheless, it remains as a piece of further research to integrate a sufficiently flexible means of combining attribute-value assignment preferences into the existing internally and externally dynamic constraint satisfaction techniques. This is in order to maximise the utilisation of available information about the preferences of modelling choices, as preference combination methods different from that using the standard min-max operations seem to be potentially more beneficial.

Although the proposed compositional modeller is intended to be used interactively, it needs to select an appropriate default model. In this respect, the current implementation can take formal requirements, such as which variables must be exogenous or endogenous, and allows preferences to be assigned to individual assumptions. Automating the selection of these requirements and preferences based on the presumed needs of the user forms a fascinating, yet challenging Also, the important issues of space and time complexity of the proposed approach to compositional modelling remains to be studied in detail, although the work presented in (Miguel, I. & Shen, Q. 2000) has demonstrated that the solution techniques to DCSPs are efficient. Once the task of compositional modelling with disaggregations is translated into a DCSP, computational complexity can be coped with by the use of these efficient algorithms.

Acknowledgements

The first author is supported by a scholarship of the Faculty of Science and Engineering, University of Edinburgh. Both authors are grateful to Ian Miguel and Robert Muetzelfeldt for helpful discussions.

References

Dechter, R., and Dechter, A. 1988. Belief maintenance in dynamic constraint networks. In *Proceedings of the 7th National Conference on Artificial Intelligence*, 37–42.

Falkenhainer, B., and Forbus, K.D. 1991. Compositional modeling: finding the right model for the job. *Artificial Intelligence* 51:95–143.

Heller, U., and Struss, P. 1998. Diagnosis and therapy recognition for ecosystems - usage of model-based diagnosis techniques. In *Proceedings of the 12th International Symposium "Computer Science for Environment Protection"*.

Holling, C. 1959. Some characteristics of simple types of predation and parasitism. *Canadian Entomologist* 91:385–398.

Keppens, J., and Shen, Q. 2000. Towards compositional modelling of ecological systems via dynamic flexible constraint satisfaction. In *Proceedings of the 14th International Workshop on Qualitative Reasoning about Physical Systems*, 74–82.

Keppens, J., and Shen, Q. 2001. On compositional modelling. To appear in Knowledge Engineering Review.

Miguel, I., and Shen, Q. 2000. Dynamic flexible constraint satisfaction. *Applied Intelligence* 13(3):231–245.

Mittal, S., and Falkenhainer, B. 1990. Dynamic constraint satisfaction problems. In *Proceedings of the 8th National Conference on Artificial Intelligence*, 25–32.

Robertson, D.; Bundy, A.; Muetzelfeldt, R.; Haggith, M.; and Uschold, M. 1991. *Eco-Logic. Logic-Based Approaches to Ecological Modelling.* MIT Press.

Struss, P. 1998. Artificial intelligence for nature - why knowledge representation and problem solving should play a key role in environmental decision support. In Haasis, H.D., and Ranze, K.C., eds., *Computer Science for the Environmental Protection '98.* Metropolis Verlag.

Verfaillie, G., and Schiex, T. 1994. Solution reuse in dynamic constraint satisfaction problems. In *Proceedings of the 12th National Conference on Artificial Intelligence*, 307–312.

Verhulst, P. 1838. Recherches mathématiques sur la loi d'accroissement de la population. *Nouveaux mémoires de l'académie royale des sciences et belles-lettres de Bruxelles* 18:1–38.