Granularity and disaggregation in compositional modelling with applications to ecological systems

Keppens, Jeroen; Shen, Qiang

Published in:
Applied Intelligence

DOI:
10.1007/s10489-006-0107-y

Publication date:
2006

Citation for published version (APA):
Keppens, J., & Shen, Q. (2006). Granularity and disaggregation in compositional modelling with applications to ecological systems. *Applied Intelligence*, *25*(3), 269-292. https://doi.org/10.1007/s10489-006-0107-y

Granularity and Disaggregation in Compositional Modelling with Applications to Ecological Systems

Jeroen Keppens[a] and Qiang Shen[b]
[a] Dept. of Computer Science, King's College London, Strand, London WC2R 2LS, UK
[b] Dept. of Computer Science, University of Wales, Aberystwyth SY23 3DB, UK

Abstract

In the past decade, compositional modelling (CM) has established itself as the predominant knowledge-based approach to construct mathematical (simulation) models automatically. Although it is mainly applied to physical systems, there is a growing interest in applying CM to other domains, such as ecological and socio-economic systems. Inspired by this observation, this paper presents a method for extending the conventional CM techniques to suit systems that are fundamentally presented by interacting populations of individuals instead of physical components or processes. The work supports building model repositories for such systems, especially in addressing the most critical outstanding issues of granularity and disaggregation in ecological systems modelling.

Keywords: model-based reasoning, compositional modelling, granularity, disaggregation, ecological systems.

1 Introduction

Compositional modelling techniques [9, 20] offer tools for explicit knowledge representation and facilitating model based reasoning [14, 22]. Although conventionally designed for engineered physical systems, compositional modellers have been recently applied to other domains, such as ecological [21, 17, 30, 32] and socio-economic [3] modelling. Compositional model repositories are compositional modellers aimed at storing established domain modelling knowledge and composing models of systems with a given configuration, in an efficient and effective manner [21]. This paper presents work following this line of development for ecological systems.

The main difference between engineered physical and ecological systems is that the latter consists of interacting populations of individuals, whose specific behaviour is difficult to predict, whereas the former consists of interacting components that have been designed with determined behaviour. Consequently, the behaviour of ecological systems is very difficult to formalise accurately. Nevertheless, from as early as the late 18th century, studies [26] have discovered that populations tend to behave according to certain patterns. Although the mathematical descriptions of such patterns are only a crude approximation of the dynamics of ecological systems, they are a useful tool in the monitoring and control of this type of system [11, 12].

One way to improve the quality of an ecological system model is to refine the classification of individuals in populations. This is a practical approach whenever certain phenomena have a different effect upon identifiable parts of the population involved. For example, a popular mathematical description of the growth phenomenon of human populations is the Malthusian principle, which states that the number of births B is proportional to the population size N:

$$B = b \times N$$

where b is the birth rate specific to the population under consideration. This description may be too coarse to answer certain questions. For instance, humans aged 20 to 30 have a much greater impact on the birth rate than humans aged 40 to 50. Thus, the Malthusian growth model could be expanded to:

$$B = \sum_i b_i \times N_i$$

where b_i and N_i are the birthrates and population sizes specific to the parts of the population that correspond to a partition of the age range.

Choosing the granularity at which populations are described in a model is a problem of particular interest in ecological systems modelling. While it is similar to the *decomposition* of physical system components (for instance, describing the behaviour of a pipe by means of a sequence of resistors and capacitors), there is a crucial difference that makes the application of conventional decomposition techniques inefficient. Decomposition is the partitioning of individuals into a set of functionally distinct component-individuals, and therefore, results in an entirely different set of individuals. For example, the decomposition of a pipe may result in a set of resistors and capacitors instead of a single pipe itself. Disaggregation is the partitioning of a population of individuals into a set of subpopulations of the original population, and therefore, it is an operation that results in outputs that are structurally equivalent to the input. For example, the disaggregation of a population of humans according to age ranges results in more populations of humans. Different types of disaggregation of the same type of population can therefore be combined with one another in any order. In other words, if there are n attributes (e.g. age, gender, social class, income, race, etc.) with respect to whom a population can be disaggregated, then that population can, in principle, be disaggregated according to each of the $2^n - n - 1$ combinations of more than one of these attributes as well.

Automated modellers normally implement granularity decisions by committing to component models that are associated with a certain choice of grain. Some approaches employ *hierarchical representations* to describe the behaviour of components or processes at different levels of detail [9, 29]. Other approaches rely on a ranking of descriptions of each process' or component's behaviour, in terms of complexity [25, 28, 31]. Therefore, an implementation of disaggregation with an automated modelling methodology that solely relies on model fragments [9] or model templates [31] would involve building templates for each combination of attributes according to which a population may be disaggregated. Thus, if disaggregations of a class of populations with respect to n distinct attributes are considered, the aforementioned decomposition approaches would require 2^n templates in order to describe all relevant forms of disaggregation the population class. This renders conventional decomposition approaches inefficient for disaggregating populations.

Inspired by the above observation, this paper presents knowledge based methods for adapting population granularity in automated construction of models of ecological systems. Section 2 describes the background of compositional model repositories. A novel approach, called disaggregation, is introduced in Section 3. Finally, Section 4 concludes the paper. Throughout the paper, theoretical discussions are illustrated with examples from the population dynamics domain.

2 Background

2.1 Compositional model repositories

Given a scenario describing the configuration of the system under consideration, the objective of a compositional model repository is to construct a mathematical model, called scenario model, that describes the behaviour of that system, meeting certain formal criteria. These criteria are instances of predefined properties of the model (e.g. the scenario model describes the predation of a given population or the scenario model contains an endogenous variable describing the size of a given population). To construct scenario models, a compositional model repository employs a knowledge base containing partial models of the processes that may occur in the systems of interest, and it utilises certain inference mechanisms to instantiate and compose these partial models with respect to a given scenario. Note that the notion of partial model herein refers to parts of models, such as individual equations or parts of individual equations, but not to the concept of partial model as defined in the Modelica language [33]

As such, compositional model repositories are particularly useful to answer queries about the behaviour of a system for which little or no data is available, but domain specific methodologies to describe certain types of behaviour are. This situation occurs quite frequently in the ecological modelling domain, where data collection is always expensive and often impossible. Consider, for example, the problem estimating the impact of introducing a new species into an existing ecosystem. The new species will interact with the existing ones as it may prey on, compete with and be preyed on by other species. Collecting data on these interactions in the given environment is not feasible without actually introducing the new species and suffering the actual impact. It is possible, however, to generate a set of potentially suitable models of the new environment, run simulations under different hypothetical conditions, and analyse the broad behavioural patterns that can be observed in the different simulations. Compositional model repositories aim to assist in this task by facilitating automated construction of system models.

The results presented in this paper are only claimed to be applicable to deductive compositional modellers, which aim at deriving a mathematical model from a structural description of the system and include compositional model repositories. It is worth noting, however that in addition to deductive compositional modellers, there are a wide variety of other types

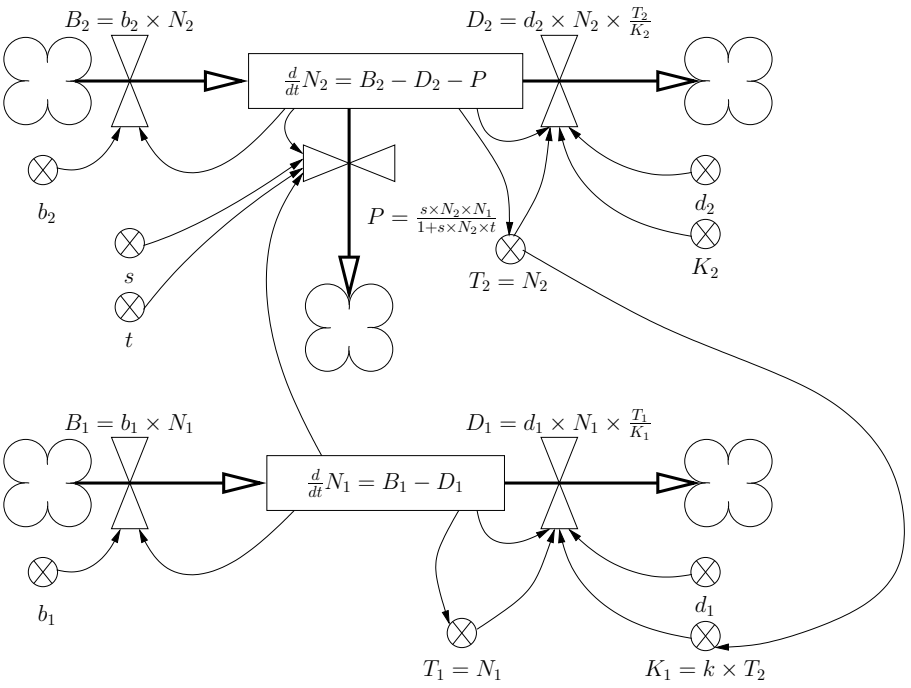


Figure 1: Behavioural model of the two species predation scenario

of automated modeller that tackle different problems. For example, research into system identification has resulted in inductive modellers that learn models from observed behaviour [2, 23, 34]. Another important strand of related work has been done in the automated diagnosis domain. It includes abductive compositional modellers [16] that search for a structural scenario for a system whose predicted behaviour matches observed behaviour. Readers interested in a detailed discussion of the tasks solved by other types of compositional modeller and how they are related to automated modelling approaches in general are referred to [20] for a detailed overview.

2.2 Scenarios and scenario models

Definition 1 A *model* M 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 [24]) of only real-world objects and interrelationships is called a *scenario*. A typical example of a scenario $\langle P_s, R_s \rangle$ studied in ecology textbooks is

```
(defScenario pred-prey-scenario
  :participants ((predator :type population)
                 (prey :type population))
  :relations ((feeds_on predator prey)))
```

where,

$$P_s = \{\texttt{predator}, \texttt{prey}\}$$
$$R_s = \{\texttt{(feeds\_on predator prey)}\}$$

A representation of variables and equations formalising or approximating the behaviour of the system is called a *scenario model*. For example, Figure 1 shows a mathematical model of the aforementioned scenario that combines the logistic [36]

and Holling [18] models and that employs the system dynamics stock-flow formalism [11]. The two squares in this figure are stocks representing the population sizes of `predator` and `prey`. These concepts are denoted by the variables N_1 and N_2 respectively. The thick arrows represent inflows into and outflows from those stocks. There are three types of flows: (1) births of new individuals belonging `predator` and `prey` (B_1 and B_2), (2) natural deaths of existing individuals belonging `predator` and `prey` (D_1 and D_2), (3) and deaths of `prey` due to predation by `predator` (P). Note that these six variables all represent real numbers, as population size is conventionally described in ecological models by the combined biomass of all individuals in the population.

The small circles represent other variables employed in the equations and the small arrows correspond to influences between variables. These variables and influences describe the population dynamics processes acting upon the `predator` and `prey` populations. They are modelled as follows:

- Changes in population size are formalised by means of the logistic population growth model [36]. This type of model presumes that births are proportional to population size (i.e. $B_i = b_i \times N_i$) and that deaths are proportional to population size and increase as the population size is closer to a theoretical maximum K_i (i.e. $D = d_i \times N_i \times \frac{T_i}{K_i}$), where $T_i = N_1$.
- Predation is formalised by means of the Holling predation model [18]. In this approach, deaths of `prey` due to predation is described by Holling's disc equation: $\frac{s \times N_1 \times N_2}{1+s \times N_2 \times t}$, where s denotes the search rate, or the proportional hunting area a predator can search per time unit spent on searching, and where t denotes the number of time units required to handle a single prey (e.g. eating, digesting, etc.). The sustenance provided by the `prey` population to the `predator` population is incorporated by setting the theoretical maximal population as proportional to the number of available prey (i.e. $K_1 = k \times T_2$).

More formally, Figure 1 depicts a model $\langle P_m, R_m \rangle$ (in the sense of definition 1, where:

$$
\begin{aligned}
P_m = \{&N_1, B_1, D_1, T_1, K_1, b_1, d_1, N_2, B_2, D_2, T_2, K_2, b_2, d_2, P, s, t, k\} \\
R_m = \{&\frac{d}{dt} N_1 = B_1 - D_1, \quad T_1 = N_1, \\
&\frac{d}{dt} N_2 = B_2 - D_2 - P, \quad T_2 = N_1, \\
&B_1 = b_1 \times N_1, \quad B_2 = b_2 \times N_2, \\
&D_1 = d_1 \times N_1 \times \frac{T_1}{K_1}, \quad D_2 = d_2 \times N_2 \times \frac{T_2}{K_2}, \\
&P = \frac{s \times N_2 \times N_1}{1 + s \times N_2 \times t}, \quad K_1 = k \times T_2\}
\end{aligned}
$$

This is not the only feasible scenario model. In ecological modelling textbooks, e.g. [15], a variety of population growth and predation models are typically presented. However, these models of individual phenomena can be composed into models describing the combined effect of different phenomena. Therefore, compositional model repositories employ a knowledge base that contains descriptions of partial models, called *model fragments* [9, 20], and they possess inference mechanisms that search for ways to combine these partial models with regard to a given scenario. The partial models described by the model fragments can not be composed arbitrarily, however. Restrictions, called *inconsistencies*, are imposed over the composition of model fragments by the representation formalism adopted and the model properties required of the scenario model.

2.3 Model space

The model space is a representation containing sufficient information to derive all possible models describing plausible behaviour of the given scenario. The remainder of this paper will therefore focus on this significant component of the compositional model repository (CMR). For a detailed discussion on how the model space is constructed and on how appropriate scenario models are derived from the model space, the reader is referred to [21].

2.3.1 Assumption-based truth maintenance

An assumption-based truth maintenance system (ATMS) is mechanism that efficiently records how each piece of inferred information depends on presumed information and facts and how inconsistencies have arisen. This section summarises

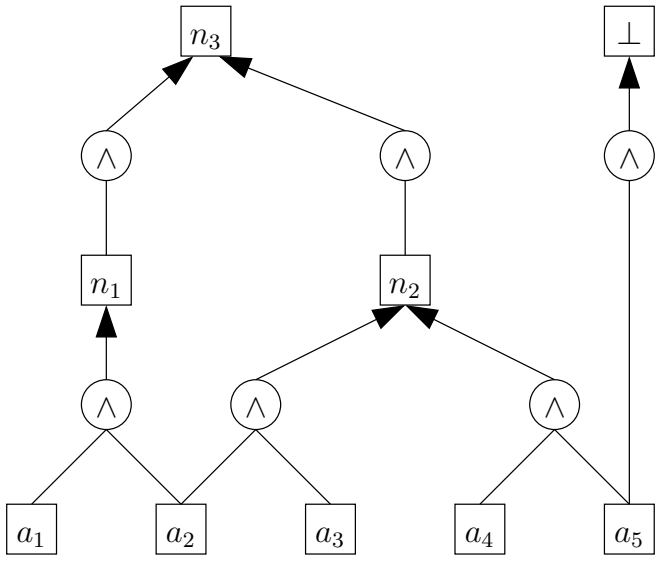


Figure 2: Sample ATMS

the functionality of an ATMS as it is extensively employed by this work. For more details, the reader is referred to the original papers [6, 7].

In an ATMS, each piece of information of relevance to the problem solver is stored as a *node*. Some pieces of information are not known to be true and cannot yet be inferred from other pieces of information. The plausibility of these is determined through the inferences made from them, and they are represented by a special type of node, called *assumption*. Figure 2 is a graphical representation of a sample ATMS with 3 nodes (n_1, n_2, n_3) and 5 assumptions ($a_1, \ldots, a_5$).

Inferences between pieces of information are maintained within the ATMS as inferences between the corresponding nodes. The ATMS can take inferences, called *justifications* of the form $n_i \wedge \ldots \wedge n_j \rightarrow n_m$, where $n_i, \ldots, n_j, n_m$ are nodes (possibly including assumptions) representing things that the problem solver is interested in. The sample ATMS of Figure 2 contains the following justifications:

$a_1 \wedge a_2 \rightarrow n_1 \quad n_1 \rightarrow n_3$
$a_2 \wedge a_3 \rightarrow n_2 \quad n_2 \rightarrow n_3$
$a_4 \wedge a_5 \rightarrow n_2$.

Note that some nodes may be deemed factually true. Such nodes, called *facts*, are justified by an empty justification, denoting that they are true in any possible world.

An ATMS can also take justifications, called *nogoods* that lead to an inconsistency, i.e. justifications of the form $n_i \wedge \ldots \wedge n_j \rightarrow \perp$. The latter nogood implies that at least one of the statements in $\{n_i, \ldots, n_j\}$ must be false. The sample ATMS of Figure 2 contains the following nogood: $a_5 \rightarrow \perp$.

Based on the given justifications and nogoods, the ATMS computes a *label* for each (non-assumption) node. A label is a set of *environments* and an environment is a set of assumptions. An environment A depicts a possible world where all the assumptions in A are true. The label $L(n)$ of a node n describes all possible worlds in which n can be true. For reasons of efficiency and effectiveness, the label computation algorithm of the ATMS guarantees that each label is:

- *Sound*: Each environment describes a possible world that logically entails the node. In other words, the presumption that all assumptions in an environment from the label of a node are true is a sufficient condition to derive that node.
- *Consistent*: No environment in the label of a node describes an impossible world. In other words, no environment entails the $\perp$ node.
- *Complete*: The label describes all possible worlds. In other words, if there is a consistent conjunction of assumptions that entails the node, then the set of those assumptions is either included in the label of the node, or a superset of a certain environment in the label of the node.
- *Minimal*: The label does not contain possible worlds that are less general than one of the other possible worlds it contains (i.e. environments that are supersets of any other environment in the label).

In the sample ATMS of Figure 2, the labels of the nodes are as follows:

$$L(n_1) = \{\{a_1, a_2\}\}$$
$$L(n_2) = \{\{a_2, a_3\}\}$$
$$L(n_3) = \{\{a_1, a_2\}, \{a_2, a_3\}\}$$
$$L(\bot) = \{\{a_5\}\}$$

The concepts of soundness, consistency, completeness and minimality can be illustrated by means of the the label of n_3:

- $L(n_3)$ is sound because it can be shown that n_3 follows from both environments:

$$a_1 \wedge a_2 \rightarrow n_1 \rightarrow n_3$$
$$a_2 \wedge a_3 \rightarrow n_2 \rightarrow n_3$$

- $L(n_3)$ is consistent because neither $\{a_1, a_2\}$ nor $\{a_2, a_3\}$ entails $\bot$.
- n_3 is entailed by each of the following consistent environments:

$$\{a_1, a_2\}, \{a_2, a_3\}, \{a_1, a_2, a_3\}, \{a_1, a_2, a_4\}, \{a_2, a_3, a_4\}, \{a_1, a_2, a_3, a_4\}$$

  Note that $\{a_5\}$ is an inconsistent environment because ($\{a_5\} \in L(\bot)$). Therefore, environments that entail n_3 but include a_5, such as $\{a_4, a_5\}$ are excluded from the above list of environments. As each of the environments in the above list is a superset of one of the environments in $L(n_3)$, $L(n_3)$ is indeed complete.

- Finally, $L(n_3)$ is minimal because $\{a_1, a_2\} \not\subset \{a_2, a_3\}$ and $\{a_2, a_3\} \not\subset \{a_1, a_2\}$ and the singletons $\{a_i\}, i = 1, 2, 3$ do not entail n_3.

2.3.2 Knowledge representation formalism

Participants refer to the objects of interest, including both real-world and conceptual objects, which are involved in the scenario or its model. For instance, a population of a species is a typical example of a real-world object, and a variable that expresses the number of individuals of this species forms an example of a conceptual object.

Relations describe how the participants are related to one another. As with participants, some relations represent a real-world relationship, such as: `predation(frog, insect)`; others may be conceptual in nature, such as Eq. (1), which describes the important textbook model of logistic population growth:

$$\texttt{change} = \frac{d}{dt}\texttt{size} = \texttt{change-rate} \times \texttt{size} \times (1 - \frac{\texttt{size}}{\texttt{capacity}}) \tag{1}$$

To be consistent with other compositional modelling approaches, this paper employs a LISP-style notation for relations. As such, the above two sample relations become:

```
(predation frog insect)
(== change (* change-rate size (- 1 (/ size capacity))))
```

Some relations, called *composable relations*, can be combined with other relations. These are conventionally employed in compositional modellers to describe parts of equations, such as the terms of a sum and the factors of product, in isolation [1, 21]. To avoid overcomplication, this paper only employs composable additive and composable substractive relations. They are of the form $p = C^+(e)$ and $p = C^-(e)$ respectively, where p is a participant and e is the relation, normally describing part of an equation, that can be combined with others. Any given set of composable additive and subtractive relations

$$p = C^+(e_1), \ldots, p = C^+(e_m), p = C^-(e_{m+1}), \ldots, p = C^-(e_n)$$

can be combined into a single relation of the form:

$$p = e_1 + \ldots + e_m - e_{m+1} - \ldots - e_n$$

Assumptions form a special type of relation. They are hypotheses or presumptions that can be made in the construction of a scenario model. As a scenario to be modelled does not usually provide a consistent and appropriate set of assumptions upon which to base the resulting scenario model, it is up to the compositional modeller to find such an assumption set.

The version of the implemented modeller discussed herein employs two types of assumption: relevance and model assumptions [21]. *Relevance assumptions* state what phenomena are to be included in or excluded from the scenario model. The general format of a relevance assumption is shown in Eq. (2). The phenomenon that is incorporated in the scenario model when describing a relevance assumption is identified by ⟨*name*⟩ and is specific to the subsequent participants or relations. For example, the relevance assumption Eq. (3) states that the growth of participant `?population` is included in the model.

$$\texttt{(relevant}\ \ \langle \textit{name} \rangle\ \ [\{\langle \textit{participant} \rangle\} \mid \langle \textit{relation} \rangle]\texttt{)} \tag{2}$$

$$\texttt{(relevant growth ?population)} \tag{3}$$

Model assumptions specify which type of model is utilised to describe the behaviour of a certain participant or relation. The formal specification of a model assumption is given in Eq. (4). Often the ⟨*name*⟩ in Eq. (4) corresponds to the name of a known (partial) model of the phenomenon or process being described. The example in Eq. (5) states that the population `?population` is being modelled using the logistic approach.

$$\texttt{(model}\ \ [\langle \textit{participant} \rangle \mid \langle \textit{relation} \rangle]\ \ \langle \textit{name} \rangle\texttt{)} \tag{4}$$

$$\texttt{(model ?population logistic)} \tag{5}$$

2.3.3 Model Space composition

Using the aforementioned representational concepts, a model space is constructed by instantiating modelling knowledge from the knowledge base into an ATMS. To this end, the knowledge base contains different types of rules, which will herein be named model construction rules and will be briefly described below. As the actual procedure that accomplishes this has been presented in detail in an earlier paper [21], this section focuses on contents of the model space itself rather than on the way it is generated, in order to keep the paper self-contained.

In the ATMS representing the model space, each node contains a participant, a relation or an instantiated model construction rule. The assumption nodes correspond to model design decisions formalised by relevance and model assumptions. The participants and relations from the given scenario are stored by facts in the scenario space. All other (synthesised) participants and relations logically follow from the given participants and relations, and certain assumptions nodes and instantiated model construction rules, as described in the model space.

Figure 3 demonstrates these ideas by showing a portion of a possible model space. The geometric shapes in this figure correspond to nodes in the model space while the arcs represent justifications. In particular, rounded rectangles are associated with assumptions, diamonds with applied model construction rules and other rectangles with nodes containing a participant or relation.

The three (non-rounded) rectangular root nodes in this scenario space correspond to participants and relation of the sample scenario shown in Section 2.2. The remainder of the scenario space shows how further participants and relations can be inferred from the given scenario and certain assumption and instantiated model construction rules. For instance, given the population named `predator`, assuming that the growth of the `predator` population is relevant, and by applying a certain model construction rule named μ_1, the model construction algorithm incorporates four new participants N_1, B_1, D_1 and T_1 and seven new relations `size-of(`N_1`,predator)`, `births-of(`B_1`,predator)`, `deaths-of(`D_1`,predator)`, `total-of(`T_1`,predator)`, $\frac{d}{dt}N_1 = C^+(B_1)$, $\frac{d}{dt}N_1 = C^-(D_1)$, and $T_1 = C^+(N_1)$.

The remainder of the partial model space given in Figure 3 describes the logistic growth model for the `predator` population, the growth phenomenon for the `prey` population, the logistic growth model for the `prey` population, the phenomenon of predation of the `prey` population by the `predator` population, and the holling predation model associated with the latter phenomenon.

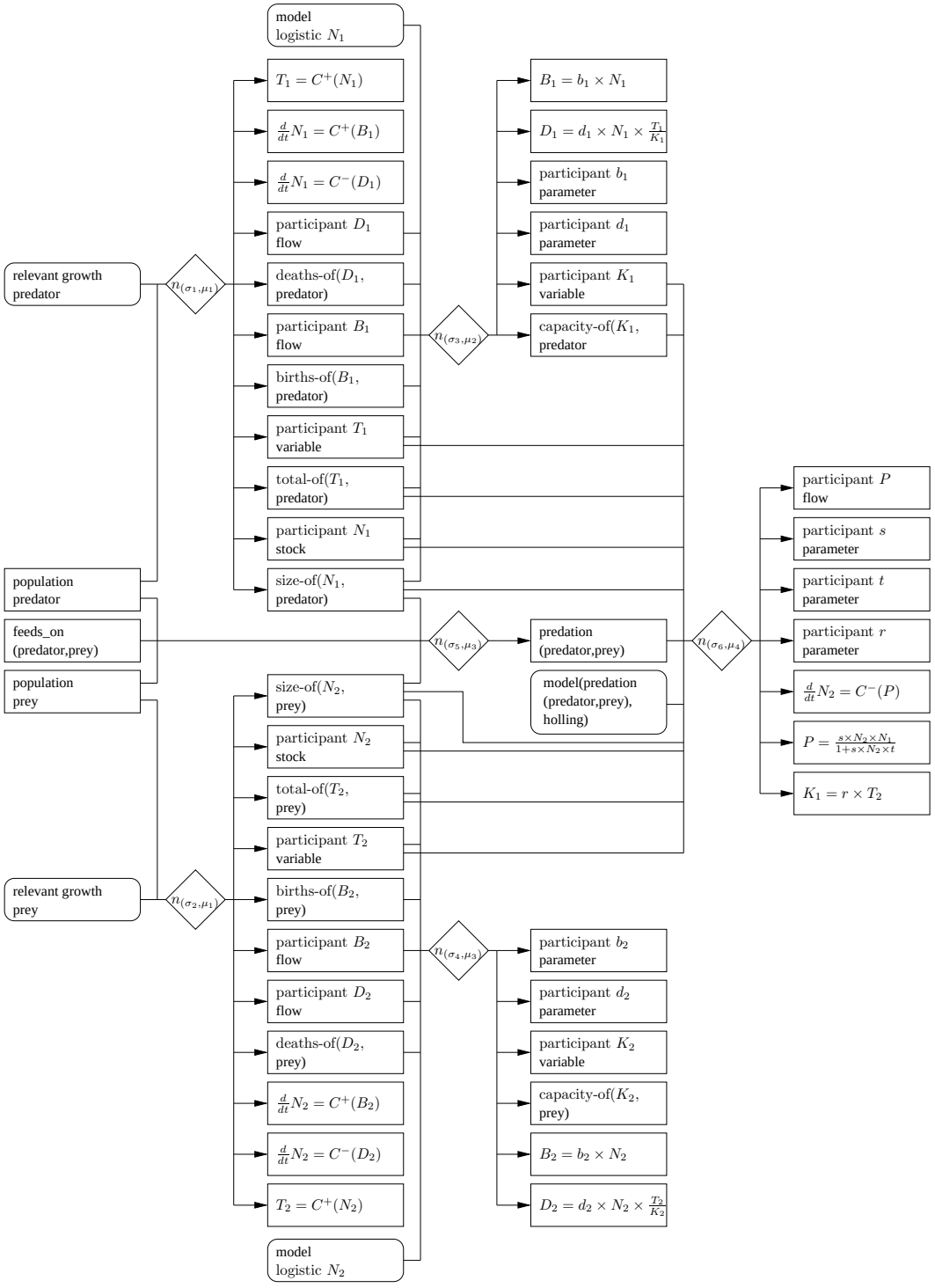


Figure 3: Partial sample model space

There are a number of different types of model construction rule that provide generic domain knowledge with which the model space is constructed. *Model fragments* are the most important type of model construction rule as they describe the conditions and assumptions under which the participants and relations that constitute a scenario model are valid. Their use is best explained by means of an example. The partial model space of Figure 3 contains six applications of model fragments. Each of these is denoted by a node $n_{(\sigma,\mu)}$, where μ refers to the applied model fragment and σ to the substitution required to map the participants and relations within the μ onto their instances in the model space. Consider for example, $n_{(\sigma_1,\mu_1)}$, which is the model rule application described above. Here, μ_1 represents the following model fragment:

```
(defModelFragment population-growth
  :source-participants ((?p :type population))
  :assumptions ((relevant growth ?p))
  :target-participants ((?pn :type stock)
                        (?pb :type flow)
                        (?pd :type flow)
                        (?t  :type variable))
  :postconditions ((size-of ?p-size ?p)
                   (births-of ?pb ?p)
                   (deaths-of ?pd ?p)
                   (total-of ?t ?p)
                   (d/dt ?pn (C+ ?pb))
                   (d/dt ?pn (C- ?pd))
                   (== ?t (C+ ?pn))))
```

and σ_1 is the substitution:

$$\sigma_1 = \{\texttt{?p}/\texttt{predator}, \texttt{?p1}/N_1, \texttt{?pb}/B_1, \texttt{?pd}/D_1, \texttt{?t}/T_1\}$$

Other types of model construction rules include *property definitions*, which describe the conditions under which a particular property is satisfied by a scenario model, and *inconsistencies*, which describe conditions under which a scenario model is not consistent. Their application follows the same procedure as the one employed for model fragments.

2.4 Model selection and the problem solver

The ultimate objective of a compositional model repository is to derive one or more scenario models that describe the behaviour of a given scenario, that are consistent, and that satisfy certain predefined properties. This is achieved in two phases. First, one or more sets of assumptions $A_i, i = 1, \ldots, n$ that entail a scenario model are searched. Then, the scenario models that follow from the discovered assumption sets $A_i, i = 1, \ldots, n$ are constructed by extracting, for each A_i, the participants and relations that A_i entails in the scenario space (and combining those extracted relations that are composable).

Let S denote the facts and justifications that constitute the scenario, P be a set of properties that the resulting scenario model must satisfy, and $\mathcal{J}$ be the justifications in the scenario space. Then, any set A_i of assumptions entails a scenario model if (i) A_i is consistent ($A_i, S, \mathcal{J} \nvdash \bot$), (ii) A_i entails all the properties in P ($\forall p \in P, (A_i, S, \mathcal{J} \vdash p)$), and (iii) A_i includes no relevance or model assumption containing a participant or relation that is not entailed by A_i ($\forall$ `(relevant` $n\ X$`)` $\in A_i, (A_i, S, \mathcal{J} \vdash X)$ and $\forall$ `(model` $X\ n$`)` $\in A_i, (A_i, S, \mathcal{J} \vdash X)$). The problem of finding sets of assumptions that meet the above three criteria can be easily accomplished by means of ATMS-based constraint solvers, such as those discussed in [8, 27]. Recent work presented in [21] has shown an extended version of such a constraint solver with an application to compositional model repositories.

For each assumption set A_i that has been discovered by the constraint solver, a scenario model is derived by extracting all participants and relations it entails in the scenario space. That is, each node representing either a participant or relation, whose label contains a subset of A_i, is deemed to be part of the scenario model entailed by A_i, and all other nodes are not. For example, given the assumption set in the model space of Figure 3, the assumption set $A =$

$\{$`relevant(growth,predator),model(logistic,`N_1`)`$\}$ entails the scenario model whose participant set is $\{N_1, T_1, B_1, D_1, b_1, d_1, K_1\}$ and whose relation set consists of:

$\frac{d}{dt}N_1 = C^+(B_1)$	size-of(N_1, predator)
$\frac{d}{dt}N_1 = C^-(D_1)$	total-of(T_1, predator)
$T_1 = C^+(N_1)$	births-of(B_1, predator)
$B_1 = b_1 \times N_1$	deaths-of(D_1, predator)
$D_1 = d_1 \times N_1 \times \frac{T_1}{K_1}$	capacity-of(K_1, predator)

3 Disaggregation

As argued previously, methods for choosing the granularity of the components of a physical system are not suitable for specifying the granularity of populations of individuals in a mathematical model. The latter type of granularity selection involves partitioning the collection of individuals in a population into subpopulations. As such, there are multiple dimensions, which typically correspond to the attributes upon which the collections are defined, along which the granularity of a given population may be specified. Generally speaking, the granularity of a population may have to be changed with respect to any combination of these dimensions, depending on which attributes may affect the processes that must be described in more or less detail. While it is theoretically possible to employ existing compositional modelling techniques to describe the behaviour of populations with different grains, it is not practical because it would require that each component model in the knowledge base is described not by a single model fragment, but one for each *combination* of attributes according to which the population may be partitioned.

Instead, this work proposes to adapt the grain of models by transforming them. This model transformation approach can work in two forms: (i) *aggregation* consolidates two or more populations into a single population and adapts the description of the processes acting upon the original populations into processes acting upon the consolidated population; and (ii) *disaggregation* divides a population and the processes acting upon them into multiple populations with corresponding processes.

For efficiency reasons, it is desirable to limit the number of transformations required for the construction of a scenario model. Because populations have numerous attributes that can guide the partitioning of any given population and among the models that adequately describe the phenomena of interest, simpler ones are usually preferred over more complex ones [11, 28], most of the possible ways partitioning the population will be irrelevant to a given situation. Therefore, disaggregation is normally the preferred form of transformation. Furthermore, disaggregation helps to simplify maintenance of the knowledge base as it permits the model fragments to be represented in their simplest (aggregate) form. Thus, this section will extend the compositional model repository with a knowledge representation formalism, and an associated inference mechanism, to perform disaggregation in an automated fashion.

3.1 Disaggregation in practice

Figure 4 presents an extension of the predator prey model shown in Figure 1. In this extended model, the `prey` population has been partitioned into three subpopulations according to age. While this model seems complex, it is in fact, very similar to the original model as given in Figure 1.

Within Figure 4, the second index in the subscripts of some variables, such as the number 1 in $N_{2,1}$, refers to the subpopulation the variable concerned describes. Thus, while N_2 refers to the population size of `prey` in Figure 1, $N_{2,1}$ refers to the population size of the first age-group of `prey`. As such, it can be seen that the death and predation phenomena, the corresponding equations, and the influences to birth flow in the model of Figure 1 are defined for each subpopulation in Figure 4.

There are only two crucial deviations from the general pattern observed above. Firstly, two new types of variable are introduced into the extended model: M_2, M_3 and m_2, m_3. These new variables and the corresponding equations describe the migration flow between subpopulations 1 and 2 and the one between subpopulations 2 and 3. Secondly, individuals are not born into their own subpopulation, but into the subpopulation that corresponds to the first age-group (and therefore, each birth flow $B_{2,i}$ is a flow of new individuals into $N_{2,1}$ instead of $N_{2,i}$). The remainder of this work will present an approach for integrating this process of disaggregation, which is normally performed manually by a model builder, into compositional model repositories.

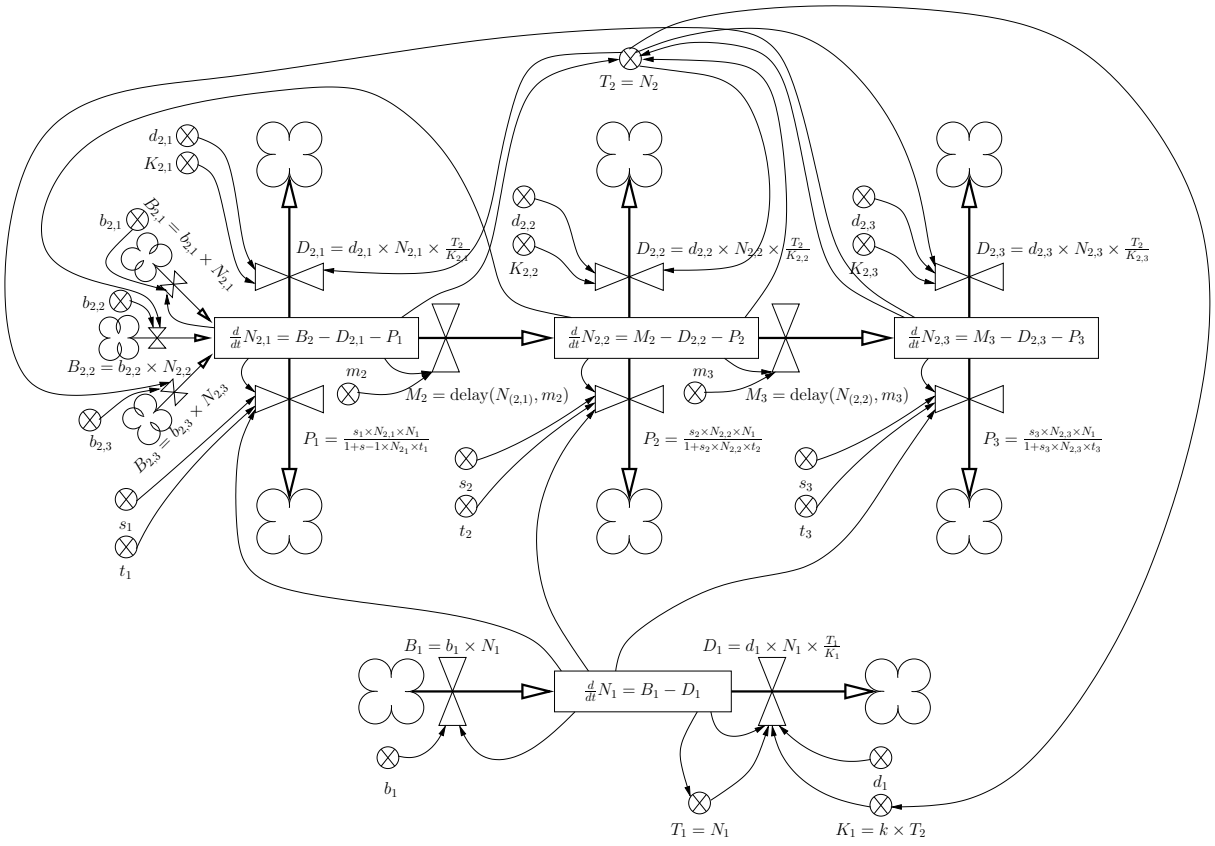


Figure 4: Holling predation model with `prey` population disaggregated in three age classes

3.2 Concept of model disaggregation

Disaggregation is the model transformation that is required if a population is partitioned into subpopulation. As mentioned before, this partitioning occurs by partition a measurable attribute of the individuals in the population, such as age, size, gender, etc. In what follows, the subpopulation will be referred to as classes, as they are subpopulations of individuals that are similar with regard to certain attributes.

In general, transforming a model, herein called the aggregate model, into a new model, herein called the disaggregate model, by disaggregating a population into n classes involves one or more of the following:

1. Replacing certain participants (referred to as participants to be disaggregated) by a set of n disaggregated participants. All other participants remain in the model as they are.
2. Replacing certain relations of the form $r(p_1^d, \ldots, p_m^d, p_1, \ldots, p_r)$ (referred to as a relation to be disaggregated), where $p_1^d, \ldots, p_m^d$ are the participants to be disaggregated, by n disaggregated relations $r(p_{1i}^d, \ldots, p_{mi}^d, p_1, \ldots, p_r)$, $i = 1, \ldots, n$. All relations describing the behaviour that remains relevant within each subpopulation can be disaggregated in this manner. Some relations, however, describe the behaviour that transcends subpopulations. When disaggregating a population into age classes, for instance, all new births produced within each age-class must be added onto the subpopulation of youngest individuals.
3. Adding additional participants and relations. These are typically used to describe migrations between the resulting disaggregate classes.

Thus, the participants in the aggregate model $\langle P, R\rangle$ can be partitioned into those that are to be disaggregated (P^a) and those that are not (P^c), with $P = P^a \cup P^c$. Similarly, the relations in such an aggregate model can be partitioned into those that are to be disaggregated (R^a), those that are not (R^c), with $R = R^a \cup R^c$. The participants and relations in a corresponding disaggregate model $\langle P*, R*\rangle$ can be partitioned into those that are disaggregated (P^d and R^d), those that are not (P^c and R^c), and those that are new to the disaggregate model (P' and R'), with $P* = P^d \cup P^c \cup P'$ and $R* = R^d \cup R^c \cup R'$. To enable a compositional modeller to produce disaggregate models, a method is herein devised

to represent formally how a disaggregate model can be produced from an aggregate one. It is of course important to understand how the participants and relations in a disaggregate model are associated to those of the original aggregate model. These associations are formalised in the following definition:

Definition 2 A model $\langle P^d \cup P^c \cup P', R^d \cup R^c \cup R' \rangle$ is said to be a *disaggregate* one of another $\langle P^a \cup P^c, R^a \cup R^c \cup R'' \rangle$ if there exists a surjection $f_P : P^d \mapsto P^a$, such that

$$\forall p^a \in P^a, \exists \{p_1^d, \ldots, p_n^d\} \subset P^d, (f_P(p_1^d) = p^a \wedge \ldots \wedge f_P(p_n^d) = p^a) \tag{6}$$

and a surjection $f_R : \text{dom}(f_R) \mapsto \text{range}(f_R)$, with $\text{dom}(f_R) \subseteq R^d$ representing the domain of f_R and $\text{range}(f_R) \subseteq R^a$ representing the range of f_R, such that

$$R^a - \text{range}(f_R) = \{r(f_P(p_q^d), \ldots, f_P(p_r^d), p_v^c, \ldots, p_w^c) \mid r(p_q^d, \ldots, p_r^d, p_v^c, \ldots, p_w^c) \in R^d - \text{dom}(f_R)\}, \tag{7}$$

and

$$\begin{aligned} &\forall r(p_q^a, \ldots, p_r^a, p_v^c, \ldots, p_w^c) \in \text{dom}(f_R), \exists r(p_s^d, \ldots, p_t^a, p_v^c, \ldots, p_w^c) \in \text{range}(f_R), \\ &f_R(r(p_q^a, \ldots, p_r^a, p_v^c, \ldots, p_w^c)) = r(p_s^d, \ldots, p_t^a, p_v^c, \ldots, p_w^c) \end{aligned} \tag{8}$$

with $f_P(p_s^d) = p_q^a, \ldots, f_P(p_t^d) = p_r^a$.

In this definition, Eq. (6) describes how the disaggregated participants in P^d are related to the participants to be aggregated in P^a, Eqs. (7) and (8) describe how the disaggregated relations in R^d are related to the relations to be disaggregated in R^a. The relations referred to in Eq. (7) are the disaggregated relations of the form $r(p_{1i}^d, \ldots, p_{mi}^d, p_1, \ldots, p_r)$ that can be mapped to aggregated relations of the form $r(p_1^d, \ldots, p_m^d, p_1, \ldots, p_r)$. In other words, they are the relations describing principles that are valid both within each subpopulation as well as to the original population as whole (before its disaggregation). The relations referred to in Eq. (8) are those that describe the behaviour that transcends subpopulations.

Consider, for example, the simple model of logistic population growth [36]:

$$\frac{d}{dt}N = C^+(B), \frac{d}{dt}N = C^-(D) \tag{9}$$

$$B = b \times N, D = d \times N \times \frac{T}{K}, T = C^+(N) \tag{10}$$

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, b 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:

$$\frac{d}{dt}N_0 = C^+(B_i), \frac{d}{dt}N_i = C^-(D_i) \tag{11}$$

$$B_i = b_i \times N_i, D_i = d_i \times N_i \times \frac{T}{K}, T = C^+(N_i) \tag{12}$$

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

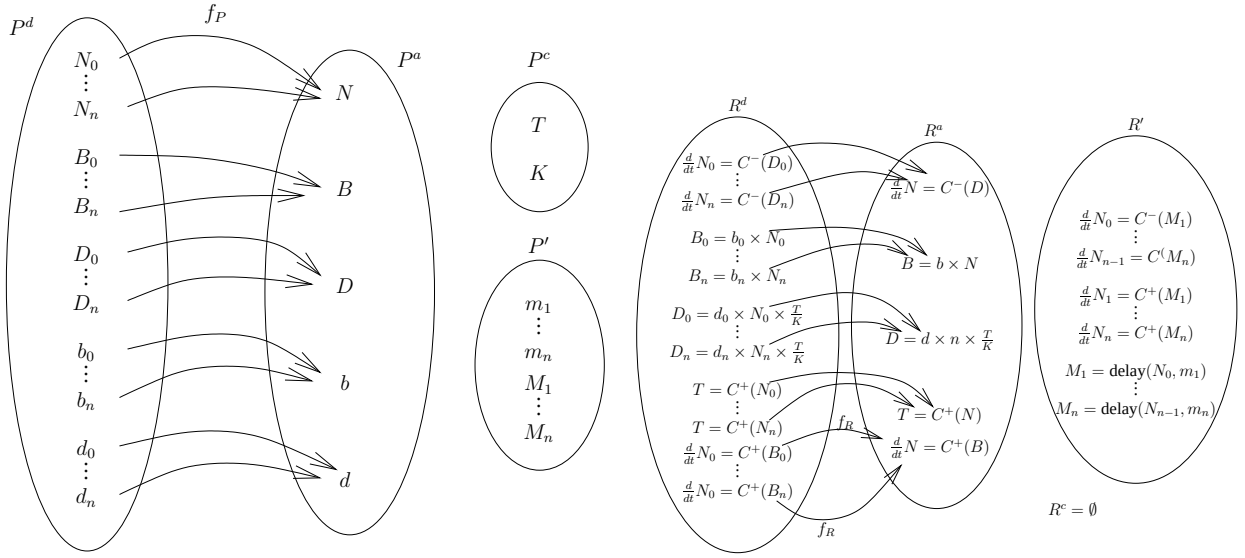
where M_i is the number of individuals that migrate from age-class $i-1$ to age-class i and m_i is the amount of time spent in age-class $i-1$.

Figure 5 shows how equations (11), (12) and (13) express a disaggregate model of equations (9) and (10). Next, a knowledge representation formalism to express these mappings is devised.

3.3 Representing disaggregation

Since a disaggregate model usually has much in common with the model it is disaggregated from, a knowledge representation, called disaggregation mapping, is introduced that makes the relations between them explicit. It specifies which participants and relations are the same, and which participants and relations are mapped to arrays of participants and relations:

Definition 3 A *disaggregation mapping* m is a tuple $\langle N, g_P, g_R \rangle$ where



(a) Participants (b) Relations

Figure 5: Disaggregating the logistic population growth model

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

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

In terms of definition 2, the participant mapping g_P states which participant classes belong to P^a and how they are mapped onto P^d of the disaggregate model. The participants that are potentially in P^a are those in the range of g_P (i.e. $p \in \text{range}(g_P)$) and they are said to be the *disaggregate participant* in m. For what follows, a transformation called the *generalised participant mapping* $g'_P : \mathcal{P} \rightarrow \mathcal{P}$ is defined such that

$$g'_P(\ldots, n_i, \ldots, p) = \begin{cases} g_P(\ldots, n_i, \ldots, p) & \text{if } p \in \text{dom}(g_P) \\ p & \text{if } p \in \mathcal{P} - \text{dom}(g_P) \end{cases}$$

The bijection g_R describes how the relations of R^a in definition 2 that do not disaggregate according to the defined classes map onto R^d. As with g_P, a transformation called the *generalised relation mapping* $g'_R : \mathcal{R} \rightarrow \mathcal{R}$ is defined such that

$$g'_R(\ldots, n_i, \ldots, r) = \begin{cases} g_R(\ldots, n_i, \ldots, r) & \text{if } r \in \text{dom}(g_R) \\ r & \text{if } r \in \mathcal{R} - \text{dom}(g_R) \end{cases}$$

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

$$g_P(i, N) = N_i, g_P(i, B) = B_i, g_P(i, D) = D_i,$$
$$g_P(i, b) = b_i, g_P(i, d) = d_i,$$
$$g_R(i, (\frac{d}{dt}N = C^+(B))) = (\frac{d}{dt}N_0 = C^+(B_i))$$
$$g_R(i, (\frac{d}{dt}N = C^-(D))) = (\frac{d}{dt}N_i = C^-(D_i))$$
$$g_R(i, (B = b \times N)) = (B_i = b_i \times N_i)$$
$$g_R(i, (D = d \times N \times \frac{T}{K})) = (D_i = d_i \times N_i \times \frac{T}{K})$$

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 from $\frac{d}{dt}N = C^+(B)$ is given in Eq. (9) onto $\frac{d}{dt}N_0 = C^+(B)$ as in Eq. (11).

By themselves, disaggregation mappings contain insufficient information to enable the construction of disaggregate models from a knowledge base of model fragments that are not disaggregated. As indicated earlier, in addition to the participants and relations that are mapped onto themselves or an array of similar participants and relations, disaggregation may involve the creation of new participants and relation. In the ongoing example application of disaggregation into age-classes, the disaggregate model contains a number of participants and relations to explain the migration from one age-class to the next. These participants and relations are specific to the age-class disaggregate model and are to be introduced when disaggregation is applied. The compound construct containing this additional information and the disaggregation mapping is called a disaggregation fragment. More formally:

Definition 4 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 a set of assumptions that are different from those specified 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 $P^d = \{p^d \mid p^d = g'_P(n_1, \ldots, n_l, p^a), p^a \in P^a\} \cup P^t$ and $R^d = \{r(g'_P(n_1, \ldots, n_l, p^a_q), \ldots, g'_P(n_1, \ldots, n_l, p^a_r)) \mid \wedge(r(p^a_q, \ldots, p^a_r) \in R^a) \wedge (r(p^a_q, \ldots, p^a_r) \notin \text{dom}(g_R))\} \cup \{g_R(r(p^a_q, \ldots, p^a_r)) \mid (r(p^a_q, \ldots, p^a_r) \in \text{dom}(g_R))\} \cup \Phi^t$

The usefulness of disaggregation fragments stems from the fact that most of the processes that act upon a population also act upon subsets of that population. Consequently, a number of shortcuts can be employed in the specification of g_P and g_R. The sample disaggregation fragment below illustrates this, specifying the disaggregation of 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 `?pn`, `?pb` and `?pd` respectively stand for the size, the number of births and the number of deaths of population `?p`. Note that both the `predator` and the `prey` populations in Figure 1 can be modelled in this way.

```
(defDisaggregationFragment species-age-classes
  :source-participants ((?p :type population)
                        (?pn :type stock)
                        (?pb :type flow)
                        (?pd :type flow))
  :structural-conditions ((size-of ?pn ?p)
                          (births-of ?pb ?p)
                          (deaths-of ?pd ?p)
                          (d/dt ?pn (C+ ?pb))
                          (d/dt ?pn (C- ?pd)))
  :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
                                     :scope always)
    (?pb* :type (age-classes flow)   :mapped-from ?pb
                                     :scope always)
    (?pd* :type (age-classes flow)   :mapped-from ?pd
                                     :scope always)
```

```
  (? :type (age-classes flow) :mapped-from (? :type flow)
                              :scope if-necessary)
  (? :type (age-classes parameter) :mapped-from (? :type parameter)
                                   :scope if-necessary)
  (?pm :type (array (1 ?n) flow))
  (?ts :type (array (0 (1- ?n)) variable)))
:postconditions
 (((for (?i 0 ?n) (d/dt (?pn 0) (C+ (?pb ?i)))
        :mapped-from (d/dt ?pn (C+ ?pb)))
  (for (?i 1 ?n) (d/dt (?pn ?i) (C- (?pm (1- ?i)))))
  (for (?i 1 ?n) (d/dt (?pn ?i) (C+ (?pm ?i))))
  (for (?i 1 ?n) (== (?pm i) (delay (?pn (1- ?i)) (?ts ?i))))))))
```

This disaggregation fragment is very similar to a model fragment, but there are distinctions that enable the specification of the disaggregation mapping. Firstly, the `:meta-participants` and `:mapping-types` statements are employed to specify the parameter(s) and transformation(s) from single participants in the aggregate model to sets of participants in disaggregate model. More specifically,

```
:meta-participants ((?n :type integer))
```

specifies an integer `?n` denoting the number of (age-)classes in the disaggregation and

```
:mapping-types (((age-classes ?t) :type (array (0 ?n) ?t)))
```

indicates that a participant of type `?t` can be mapped to an array containing `?n` + 1 instances of type `?t`.

The bijection g_P is specified via the target-participants of the disaggregation fragment:

- The modifier `:mapped-from` allows the specification of a disaggregated participant. Here, the stock `?pn` and the flows `?pb` and `?pd` are disaggregated participants in the disaggregation mapping. In a disaggregate model based on the disaggregation fragment, they are replaced by an array of participants. The wildcard symbol `?` is employed to specify the mapping of multiple participants. For example, the following states that all parameters are mapped to arrays of parameters:

  ```
  (?  :type (age-classes parameter) :mapped-from (?  :type parameter))
  ```

  The target-participants without the `:mapped-from` modifier (here `?pm` and `?ts`) specify new participants that must be added to a disaggregate model. `?pm` and `?ts` describe the migration of individuals between age-classes and they correspond to M_i and m_i (with $i = 1, \ldots, n$) in Figure 4.

- The modifier `:scope` is used in conjunction with `:mapped-from`. `:scope always` implies that the mapping is applicable to all participants that meet the specification. `:scope if-necessary` implies that the mapping is only applicable to those participants that meet the specification and are employed in a relation that contains one or more other participants with the same mapping. For example, the statement below is only applicable to some of the parameters in a model:

  ```
  (? :type (age-classes parameter) :mapped-from (? :type parameter)
                                   :scope if-necessary)
  ```

  In the sample model of Figure 1 (where Figure 3 is the corresponding model space), the above statement is applicable to parameter b_2 (because b is used in conjunction with the disaggregated parameters N_2 and B_2 in $\frac{d}{dt}N_2 = b_2 \times B_2$) but not to r (because r is only used in conjunction with T_1 and K_2, neither of which is disaggregated).

The bijection g_R is partially specified via the postconditions of the disaggregation fragment:

- The modifier `:mapped-from` works in the same way as for target-participants. Thus, the first postcondition states that the births (specified by `(d/dt ?pn (C+ ?pb))`) in the aggregate model is replaced by an `?n` + 1 flows (i.e. one for each subpopulation) to the first population size stock `(?pn 0)`:

```
(d/dt (?pn 0) (C+ (?pb 1)))
                ⋮
(d/dt (?pn 0) (C+ (?pb ?n)))
```

- The next two postconditions state the new relations that must be added due to the disaggregation, specifying the migrations between age classes.

Each relation $r(p_k, \ldots, p_l)$ that is not mentioned in the postconditions of the disaggregation fragment is mapped as follows:

$$g_R(i, \ldots, j, r(p_k, \ldots, p_l)) = r(g_P(i, \ldots, j, p_k), \ldots, g_P(i, \ldots, j, p_l)) \tag{14}$$

For example, $B = b \times N$ is mapped such that:

$$g_R(x, (B = b \times N))) = (B_x = b_x \times N_x)$$

The default mapping described in (14) reflects the aforementioned feature that most processes, and the relations describing them, are equally applicable to populations and subsets of these populations.

3.4 Inferring disaggregations

When applying the disaggregation fragment `species-age-classes`, specified above, certain 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:

- First, the part of the model space to which the disaggregation fragment is applicable must be identified.
- Then, the criteria for the disaggregation mapping in the disaggregation fragment must be made explicit for that part of the model space.
- Finally, the relevant part of the model space is duplicated and the disaggregation mapping is applied to it.

By means of this extended model space, disaggregate models can be composed. This section describes each of these steps and illustrates them using the sample model space of Figure 3 and disaggregation fragment `species-age-classes`.

3.4.1 Application of a disaggregation fragment

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.

Definition 5 Given a disaggregation fragment δ with $P^s = \{p_1^s, \ldots, p_m^s\}$ and $\Phi^s = \{\phi_1^s, \ldots, \phi_v^s\}$, a set of participants instances $O = \{o_1, \ldots, o_m\}$, a set of instantiated relations $R = \{r_1, \ldots, r_v\}$, a substitution $\sigma = \{p_1^s/o_1, \ldots, p_m^s/o_m\}$ and a model space Δ such that

$$(\forall o \in O, o \in \Delta) \wedge (\forall r \in R, r \in \Delta) \wedge (\forall i = 1, \ldots, v, \sigma\phi_i^s = r_i)$$

The set of nodes to which the disaggregation fragment is *applicable* with respect to a substitution σ is the set of nodes M such that

$$\Delta, M \vdash o_1, \ldots, o_m, r_1, \ldots, r_v$$
$$\neg\exists M' \subset M\Delta, M' \vdash o_1, \ldots, o_m, r_1, \ldots, r_v$$

In other words, the set of nodes to which a disaggregation fragment is applicable is the smallest set of nodes which justify the nodes that match the source-participants and structural conditions of the disaggregation fragment. The algorithm APPLYDF(Δ, d, M, σ) describes the procedure of extending and transforming the model space Δ for an application of a disaggregation fragment d, when it is deemed applicable to a set of nodes M with respect to a substitution σ. But before it can be applied, the disaggregation mapping must be fully specified.

In Figure 3, for instance, the disaggregation fragment `species-age-classes` is applicable to

- $\{n_{(\sigma_1,\mu_1)}\}$ because the set of participants $O = \{\texttt{predator}, N_1, B_1, D_1\}$ matches the source-participants of the disaggregation fragment and the set of relations $R = \{(\frac{d}{dt}N_1 = C^+(B_1)), (\frac{d}{dt}N_1 = C^-(D_1))\}$
- $\{n_{(\sigma_2,\mu_1)}\}$ because the set of participants $O = \{\texttt{prey}, N_2, B_2, D_2\}$ matches the source-participants of the disaggregation fragment and the set of relations $R = \{(\frac{d}{dt}N_2 = C^+(B_2)), (\frac{d}{dt}N_2 = C^-(D_2))\}$

To follow this theme, the following discussion will be supported by applying `species-age-classes` to $\{n_{(\sigma_2,\mu_1)}\}$.

3.4.2 Generating the disaggregation mapping

Recall that disaggregation fragments specify a set of criteria that the disaggregation mapping should satisfy in order to make it more generally applicable. When the disaggregation fragment is applied, the criteria specifying the disaggregation mapping must be applied to the relevant part of the model space, thus making g_P and g_R explicit.

Making g_P explicit involves determining which participants are to be disaggregated in the disaggregation mapping, and which ones are not. For all disaggregate participants p in the mapping, g_P is defined as

$$\begin{aligned} g_P(i, \ldots, j, p) = p_{i,\ldots,j} \text{ with } i &= 1, \ldots, n_1 \\ &\vdots \\ j &= 1, \ldots, n_l \end{aligned}$$

As defined above, some criteria are applicable to all participants that meet them, and some criteria are only applicable if they also occur in a relation with a disaggregate participant. The latter issue is resolved by means of constraint propagation with the Waltz algorithm [37].

The relevant participants in the model space of Figure 3 are N_2, B_2, D_2, T_2, K_2, P, b_2, d_2, s, t, r. According to the disaggregation fragment `species-age-classes`, the stock N_2 and the flows B_2, D_2 and P are disaggregated. The parameters b_2, d_2, s, t, r are to be disaggregated if they occur in a relation with a disaggregate participant. This is the case for all parameters, except r (which only occurs in the relation $K_1 = r \times T_2$ with the non-disaggregate participants K_1 and T_2. Thus, the *generalised participant mapping* g'_P is now defined as follows, with $i = 1, \ldots, n$:

$$\begin{array}{cccc} g'_P(i, N_2) = N_{2,i} & g'_P(i, B_2) = B_{2,i} & g'_P(i, D_2) = D_{2,i} & g'_P(i, P) = P_i \\ g'_P(i, T_2) = T_2 & g'_P(i, K_2) = K_2 & g'_P(i, b_2) = b_{2,i} & g'_P(i, d_2) = d_{2,i} \\ g'_P(i, s) = s_i & g'_P(i, t) = t_i & g'_P(i, r) = r & \end{array}$$

Making g_R explicit involves two steps:

- Some mappings are specified in the disaggregation fragment. For example, as shown above, `species-age-classes` contains a postcondition stating that:

$$g_R(i, (\frac{d}{dt}N_2 = C^+(B_2))) = (\frac{d}{dt}N_{2,0} = C^+(B_{2,i})); \quad i = 1, \ldots, n$$

- All relations that are not mentioned in the disaggregation fragment undergo the default mapping (14). Thus, the explicit specification of g_R in the ongoing example is:

$$\begin{array}{cc} g_R(i, (\frac{d}{dt}N_2 = C^-(D_2))) = (\frac{d}{dt}N_{2,i} = C^-(D_{2,i})) & g_R(i, (T_2 = C^+(N_2))) = (T_2 = C^+(N_{2,i})) \\ g_R(i, (B_2 = b_2 \times N_2)) = (B_{2,i} = b_{2,i} \times N_{2,i}) & g_R(i, (D_2 = d_2 \times N_2 \times \frac{T_2}{K_2})) = (D_{2,i} = d_{2,i} \times N_{2,i} \times \frac{T_2}{K_2}) \\ g_R(i, (\frac{d}{dt}N_2 = C^-(P))) = (\frac{d}{dt}N_{2,i} = C^-(P_i)) & g_R(i, (P = \frac{s \times N_2 \times N_1}{1+s \times N_2 \times t})) = (P_i = \frac{s_i \times N_{2,i} \times N_1}{1+s_i \times N_{2,i} \times t_i}) \end{array}$$

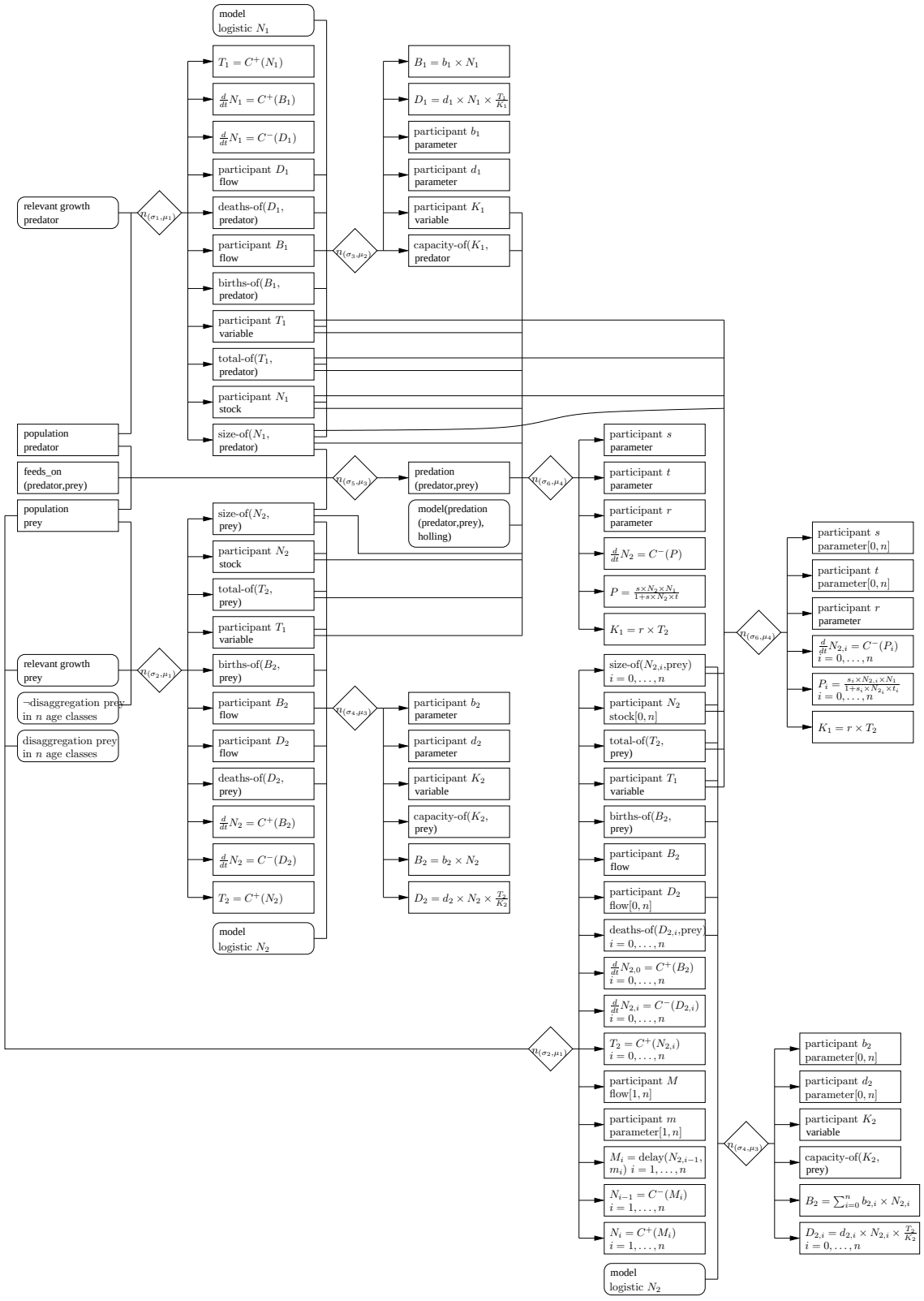


Figure 6: Model space expansion via disaggregation fragment

3.4.3 Extending the model space

The algorithm APPLYDF$(\Delta, \delta, M, \sigma)$ describes the procedure of extending and transforming the model space Δ for an application of a disaggregation fragment δ, which is applicable with respect to the set of model fragment instances M and

with a substitution σ. This algorithm works as follows:

- APPLYDF(Δ, δ, M) copies the nodes in M and those that are entailed by M in Δ, and applies the disaggregation mapping in δ to it (i.e. g'_P to the participants and g'_R to the relations). The root nodes of this transformed copy of the part of Δ are a new set of nodes that are denoted d_M inside the algorithm (or $d_M(\Delta, \delta, M)$ outside the algorithm). The nodes in d_M are justified by the justification of M conjoined with the assumptions (instantiated with substitution σ) of the disaggregation fragment A:

  $$(\wedge_{m \in M} \mathcal{J}(m)) \wedge (\sigma A)$$

  where $\mathcal{J}(m)$ represents the justification of a node m in the model space Δ.

- The justification of the original set of nodes is conjoined with the negation of the assumptions of the disaggregation fragment (instantiated with substitution σ):

  $$(\wedge_{m \in M} \mathcal{J}(m)) \wedge \neg(\sigma A)$$

In order to perform these transformations, APPLYDF$(\Delta, \delta, M, \sigma)$ maintains a queue Q of nodes that have to be copied with their consequences.

Algorithm 3.1: APPLYDF(Δ, δ, M)

$d_\theta \leftarrow \Delta$;
$d_M \leftarrow \{\}$;
$Q \leftarrow M$;
repeat
  $\mu \leftarrow$ dequeue(Q);
  $\mu' \leftarrow$ add-node$(d_\theta,$ content$(\mu))$;
  $d_M \leftarrow d_M \cup \{\mu'\}$;
  if $\mu \in M$
    then { add-justification$(d_\theta, \mu', \sigma\mathcal{J}(\mu) \wedge \sigma A)$;
              justification$(\mu) \leftarrow \mathcal{J}(\mu) \wedge \neg(\sigma A)$;
    else add-justification$(\mu', \sigma\mathcal{J}(\mu))$;
  for each $p \in$ consequents$(\mu),$ participant(p)
    do { $n \leftarrow$ add-node$(d_\theta, g'_P(\overrightarrow{i}, p))$;
             $\sigma \leftarrow \sigma \cup \{p/g'_P(\overrightarrow{i}, p)\}$;
             add-justification(d_θ, n, μ');
             for each $\mu'',$ (successor(μ'', p))
               do enqueue(Q, μ'');
  for each $r \in$ consequents$(\mu),$ relation(r)
    do { $n \leftarrow$ add-node$(d_\theta, g'_R(\overrightarrow{i}, r))$;
             $\sigma \leftarrow \sigma \cup \{r/g'_R(\overrightarrow{i}, r)\}$;
             add-justification(d_θ, n, μ');
             for each $\mu'',$ (successor(μ'', r))
               do enqueue(Q, μ'');
until $Q = \{\}$;
return $(d_\theta(\Delta, d, M) = d_\theta, d_M(\Delta, d, M) = d_M))$;

The algorithm APPLYDF(Δ, d, M, σ) returns a new, extended, model space $d_\theta(\Delta, d, M)$ which replaces the original Δ, and the set of new nodes $d_M(\Delta, d, M)$, which replaces the nodes from which the newly added parts of the model space are derived. The set of nodes $d_M(\Delta, d, M)$ corresponds to the root nodes subgraph that has been added to the original model space (this is relevant to the discussion on combining disaggregations in Section 3.5).

As discussed in Section 3.2, the application of a disaggregation fragment also involves the creation of new target-participant instances and corresponding instances of postconditions. INSTANTIATEDF(Δ, d, M, σ) is the procedure for adding these participants and relations and it returns $P^t(\Delta, d, M)$, the set of newly created participant instances, and $\Phi^t(\Delta, d, M)$, the set of newly created postcondition instances.

Algorithm 3.2: INSTANTIATEDF($\Delta, \delta, M, \sigma$)

$P \leftarrow \{\}$;
$R \leftarrow \{\}$;
for each $p \in P^t$
do $\begin{cases} \sigma \leftarrow \sigma \cup \{p/\text{gensym}()\}; \\ n \leftarrow \text{add-node}(d_\theta(\Delta, \delta, M), (\sigma p)); \\ \text{add-justification}(d_\theta(\Delta, \delta, M), n, d_M(\Delta, \delta, M)); \\ P \leftarrow P \cup \{n\}; \end{cases}$
for each $\phi \in \Phi^t$
do $\begin{cases} n \leftarrow \text{add-node}(d_\theta(\Delta, d, M), (\sigma \phi)); \\ \text{add-justification}(d_\theta(\Delta, d, M), n, d_M(\Delta, d, M)); \\ R \leftarrow R \cup \{\sigma n\}; \end{cases}$
return $(P^t(\Delta, d, M) = P, \Phi^t(\Delta, d, M) = R)$;

The application of a disaggregation fragment then consists of running algorithms APPLYDF() and INSTANTIATEDF(). The resulting model space consists of the union of the original model space, the newly disaggregated model space and the newly added participants and relations. This is formalised in Definition 6.

Definition 6 Given a model space Δ, a disaggregation fragment d and a set of nodes M (representing model fragment instances) to which d can be applied, the model space $D_\theta(\Delta, d, M)$ that is constructed by applying d to M is

$$D_\theta(\Delta, d, M) = d_\theta(\Delta, d, M) \cup P^t(\Delta, d, M) \cup \Phi^t(\Delta, d, M)$$

To continue with the ongoing example, the result of applying APPLYDF($\Delta, \delta, M, \sigma$) and INSTANTIATEDF($\Delta, \delta, M, \sigma$) to disaggregation fragment δ = `species-age-classes`, with respect to an instance of a model fragment for population growth, is shown in Figure 6. This figure is mostly similar to Figure 3 apart from the modifications outlined above:

- A new node $n_{(\sigma_7, \mu_1)}$ with the same justification as $n_{(\sigma_2, \mu_1)}$ in Figure 3, conjoined with the disaggregation assumption from `species-age-classes`. The consequences of $n_{(\sigma_7, \mu_1)}$ are those of $n_{(\sigma_2, \mu_1)}$, transformed according to g'_P and g'_R and further extended with the new participants and relations specified in the disaggregation fragment.
- The justification of the node $n_{(\sigma_2, \mu_1)}$ is extended with the negation of the disaggregation assumption from `species-age-classes`.

Finally, two important theorems can be derived from the discussion above.

Theorem 1 If a model M can be derived from a model space Δ, M can also be derived from $D_\theta(\Delta, \delta, M)$
Proof: See appendix.

Theorem 2 Given that (i) Δ is a model space, (ii) $\Delta' = d_\theta(\Delta, \delta, M)$ is the model space resulting from extending Δ by the application of a disaggregation fragment δ to a set of model fragment nodes M, (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 $A \cup A_\delta, \Delta' \vdash M_D$ is a disaggregate model of M_A
Proof: See appendix.

From theorem 1, all models that may be deduced from a model space Δ, can be deduced from the model space that is computed by APPLYDF(Δ, d, M). Further, theorem 2 shows that for each set of assumptions from which a scenario model can be deduced in a non-expanded model space with regards to a set of assumptions, a disaggregate version of that model can be deduced from the expanded model space by extending the original set of assumptions with instantiated grain assumptions taken from the disaggregation fragment.

3.5 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 participants. 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. The combined application of two disaggregation fragments d_1 and d_2 to a set of model fragment instances 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. Definition 7 formalises this concept of combined disaggregation:

Definition 7 Given two disaggregation fragments d_1 and d_2, a model space Δ and a partition $\{M_{12}, M_1, M_2, M\}$ of those nodes in Δ which represent model fragment instances, where $M_{12} \cup M_1$ and $M_{12} \cup M_2$ are minimal sets for model fragments to which d_1 and d_2 can be respectively applied, the result of the *combined disaggregation of the model space*, denoted $D_\theta(\Delta, d_2 \circ d_2, M_1 \cup M_2 \cup M_{12})$ is:

$$\begin{aligned} D_\theta(\Delta, d_2 \circ d_1, M_1 \cup M_2 \cup M_{12}) = & D_\theta(d_\theta(\Delta, d_1, M_1 \cup M_{12}), d_2, d_M(\Delta, d_1, M_{12}) \cup M_2) \cup \\ & D_\theta(d_\theta(\Delta, d_1, M_1 \cup M_{12}), d_1, P^t(d_2, M_2 \cup M_{12})) \cup \\ & D_\theta(d_\theta(\Delta, d_1, M_1 \cup M_{12}), d_1, \Phi^t(d_2, M_2 \cup M_{12})) \end{aligned}$$

Consider, for example, disaggregating a population of a particular species into q populations of subspecies. This requires a disaggregation mapping that disaggregates all participants other than T (total population) and K (maximal sustainable population). When disaggregating a population of a species according to subspecies, no new relations need to be introduced as the individuals can be presumed to remain within their subspecies for their entire lifespan. Therefore, the disaggregation fragment below specifies this transformation:

```
(defDisaggregationFragment species-subspecies
  :source-participants ((?p :type population)
                        (?pn :type stock)
                        (?pb :type flow)
                        (?pd :type flow))
  :structural-conditions ((size-of ?pn ?p)
                          (births-of ?pb ?p)
                          (deaths-of ?pd ?p)
                          (d/dt ?pn (C+ ?pb))
                          (d/dt ?pn (C- ?pd)))
  :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
                                     :scope always)
    (?pb* :type (age-classes flow)   :mapped-from ?pb
                                     :scope always)
    (?pd* :type (age-classes flow)   :mapped-from ?pd
                                     :scope always)
    (? :type (age-classes flow) :mapped-from (? :type flow)
                                :scope if-necessary)
    (? :type (age-classes parameter) :mapped-from (? :type parameter)
                                     :scope if-necessary)))
```

Now consider the combined application (see definition 7) of disaggregation into q subspecies and disaggregation into n age-classes to the logistic population growth model as described by equations (9) and (10). This involves:

1. applying the `species-subspecies` disaggregation fragment,
2. applying the `species-age-classes` disaggregation fragment, and

3. applying the `species-subspecies` disaggregation fragment to the target-participants and postconditions introduced by the application of the `species-age-classes` disaggregation fragment.

Step 1 involves applying the `species-subspecies` disaggregation fragment to the part of the model space that contains equations (9) and (10). This results in (with $j = 0, \ldots, q$):

$$\frac{d}{dt}N_j = C^+(B_j), \frac{d}{dt}N_j = C^-(D_j) \tag{15}$$

$$B_j = r_j \times N_j, D_j = d_j \times N_j \times \frac{T}{K}, T = C^+(N_j) \tag{16}$$

Step 2 involves applying the `species-age-classes` disaggregation fragment to the part of the model space that contains equations (9) and (10). This results in:

$$\frac{d}{dt}N_{0j} = C^+(B_{ij}), \frac{d}{dt}N_{ij} = C^-(D_{ij}) \tag{17}$$

$$B_{ij} = r_{ij} \times N_{ij}, D_{ij} = d_{ij} \times N_{ij} \times \frac{T}{K}, T = C^+(N_{ij}) \tag{18}$$

$$\frac{d}{dt}N_i - 1 = C^-(M_i), M_i = \text{delay}(N_i, t_i) \tag{19}$$

with $i = 0, \ldots, n$ and $j = 0, \ldots, q$ in (17) and (18) and with $i = 0, \ldots, n$ in (19).

Finally, step 3 involves applying the `species-subspecies` disaggregation fragment to the newly introduced equation (19) and results in:

$$\frac{d}{dt}N_{i-1,j} = C^-(M_{ij}), M_{ij} = \text{delay}(N_{ij}, t_{ij}) \tag{20}$$

Theorem 3 The combined disaggregation of the model space is a commutative operation. In other words,

$$D_\theta(\Delta, d_2 \circ d_1, M_1 \cup M_2 \cup M_{12}) = D_\theta(\Delta, d_1 \circ d_2, M_1 \cup M_2 \cup M_{12})$$

Proof: See appendix.

Theorem 3 ensures that the combined application of a number of disaggregation fragments yields a unique result, irrespective of the order in which the combination is implemented. Therefore, in order to combine different ways of disaggregating a model, only the individual ways of disaggregating it must be represented by disaggregation fragments. Of course, this requires that the disaggregation fragments are defined in a sufficiently general way. For example, if one disaggregation fragment introduces a new type of participant that another disaggregation fragment does not know how to process, the combined application of these disaggregation fragments can not be guaranteed to yield adequate results.

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.

4 Conclusions

This paper has addressed the issue of granularity selection in compositional model repositories applied to ecological systems. The most important distinction between ecological systems and physical systems is that the former contain interacting populations of individuals. Although existing approaches to compositional modelling offered techniques to process automatically granularity decisions with regard to individuals, they did not provide the means of granularity selection in populations.

The paper has argued that, in establishing the granularity of a population, the population can be treated as a component or a collection of individuals. Existing compositional modelling techniques are designed to deal with the decomposition of objects into to functional components. In this work a method has been devised to construct automatically scenario models

that describe the behaviour of a system containing populations, each of which be disaggregated into subpopulations. The approach employs a construct, called disaggregation fragment, to define formally possible transformations required of a scenario model in order to disaggregate one or more of the populations. Further, an algorithm has been implemented to apply the transformation described by a disaggregation fragment into a space of possible sub-models.

This work differs from other techniques in that it takes a meta-modelling approach to granularity by transforming existing models rather than replacing them by alternatives. This makes the knowledge representation more concise because it is sufficient to store only the basic, most aggregate, model fragments and legal transformations applicable to them, instead of an entire set of model fragments for each form of disaggregation. The representation is also sufficiently flexible to accommodate the composition of disaggregations. Therefore, each form of disaggregation needs to be represented only once, and all combinations of these forms of disaggregation are automatically entailed.

A number of issues were beyond the scope of this paper, however, remaining as active research. The paper has focused on automated synthesis of the general structure of models describing the behaviour of a particular ecological system. It has been assumed that the user can specify whether (s)he would like to see a population disaggregated with regard to a particular attribute of interest. The next logical step in this research involves the development of means to support or automate this decision making process. One potential approach involves supporting the elicitation of the user's preferences for a model's descriptiveness and simplicity. As reported in [21], recent work has shown the potential of allowing incorporation of such preferences into activity dynamic constraint satisfaction problem, thereby enabling the composition of those scenario models that are preferred by the user. How such work may be integrated with the present research is worth further investigation.

An important issue in the construction of disaggregate models is the selection of the number of subpopulations that a population is partitioned into with respect to a given attribute. The method introduced in this paper produces generic scenario models in which the number of subpopulations for each disaggregation remains as a parameter to be specified by the user of the emerging models. Therefore, the development of methods to decide on the values of these parameters remains as an important area of future work.

Describing the behaviour of populations instead of individuals is an important topic in model construction that does not only arise from the ecological domain. Many other disciplines, such as macroeconomics [10], resource management [4, 13, 19] and epidemiology [5, 35] are also concerned about the behaviour of populations of individuals. As the techniques introduced herein are not in principle based on any presumptions that are specific to ecological modelling, it would be interesting to investigate their applications to such related problem domains.

Acknowledgements

This work is partly supported by the UK-EPSRC grant GR/S63267. The first author has also been 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

[1] D. Bobrow, B. Falkenhainer, A. Farquhar, R. Fikes, K.D. Forbus, T.R. Gruber, Y. Iwasaki, and B.J. Kuipers. A compositional modeling language. In *Proceedings of the 10th International Workshop on Qualitative Reasoning about Physical Systems*, pages 12–21, 1996.

[2] E. Bradley, M. Easley, and R. Stolle. Reasoning about nonlinear system identification. *Artificial Intelligence*, 133:139–188, 2001.

[3] G. Brajnik and M. Lines. Qualitative modeling and simulation of socio-economic phenomena. *Journal of Artificial Societies and Social Simulation*, 1, 1998.

[4] D.W. Bunn and E.R. Larsen. Sensitivity of reserve margin to factors influencing investment in the UK electricity market. *Energy Policy*, 20(5):490–502, 1992.

[5] B. Dangerfield, Y. Fang, and C. Roberts. Model-based scenarios for the epidemiology of hiv/aids: the consequences of highly active antiretroviral therapy. *System Dynamics Review*, 17(2):119–150, 2001.

[6] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[7] J. de Kleer. A general labeling algorithm for assumption-based truth maintenance. In *Proceedings of the 7th National Conference on Artificial Intelligence*, pages 188–192, 1988.

[8] J. de Kleer. A comparison of ATMS and CSP techniques. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 290–296, 1989.

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

[10] B.S. Ferguson and Lim G.C. *Introduction to dynamic economic models*. Manchester University Press, 1998.

[11] A. Ford. *Modeling the Environment - An Introduction to System Dynamics Modeling of Environmental Systems*. Island Press, 1999.

[12] J.W. Forrester. *Principles of Systems*. Wright-Allen Press, Cambridge, MA, USA, 1968.

[13] P. Gottschalk. A system dynamics model for long range planning in a railroad. *European Journal of Operational Research*, 14(2):156–162, 1983.

[14] W. Hamscher, L. Console, and J. de Kleer, editors. *Readings in Model-Based Diagnosis*. Morgan-Kaufmann, San Mateo, 1992.

[15] A. Hastings. *Population Biology. Concepts and Models*. Springer-Verlag, New York, 1997.

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

[17] U. Heller and P. Struss. Transformation of qualitative dynamic models - application in hydro-ecology. In L. Hotz, P. Struss, and T. Guckenbienl, editors, *Intelligent Diagnosis in Industrial Applications*, pages 95–106. Shaker Verlag, 2001.

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

[19] P.T. Ittig. A model for planning ambulatory health services. *Management Science*, 24(10):1001–1010, 1976.

[20] J. Keppens and Q. Shen. On compositional modelling. *Knowledge Engineering Review*, 16(2):157–200, 2001.

[21] J. Keppens and Q. Shen. Compositional model repositories via dynamic constraint satisfaction with order-of-magnitude preferences. *Journal of Artificial Intelligence Research*, 21:499–550, 2004.

[22] B.J. Kuipers. Reasoning with qualitative models. *Artificial Intelligence*, 59:125–132, 1993.

[23] P. Langley, J. Sanchez, L. Todorovski, and S. Džeroski. Inducing process models from continuous data. In *Proceedings of the 19th International Conference on Machine Learning*, pages 347–354, 2002.

[24] R.R. Leitch, Q. Shen, G.M. Coghill, and M.J. Chantler. Choosing the right model. *IEE Proceedings on Control Theory and Applications*, 146:435–449, 1999.

[25] A.Y. Levy, Y. Iwasaki, and R. Fikes. Automated model selection for simulation based on relevance reasoning. *Artificial Intelligence*, 96:351–394, 1997.

[26] T.R. Malthus. An essay on the principle of population. Printed for J. Johnson in St. Paul's Church Yard, London, England, 1798.

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

[28] P.P. Nayak and L. Joskowicz. Efficient compositional modeling for generating causal explanations. *Artificial Intelligence*, 83:193–227, 1996.

[29] J. Rickel and B. Porter. Automated modeling of complex systems to answer prediction questions. *Artificial Intelligence*, 93:201–260, 1997.

[30] P. Salles and B. Bredeweg. A case study of collaborative modelling: building qualitative models in ecology. In Bredeweg, B., editor, *Proceedings of the International workshop on Model-based Systems and Qualitative Reasoning for Intelligent Tutoring Systems*, pages 75–84, 2002.

[31] J. Stein and L. Louca. A component-based modeling approach for system design: Theory and implementation. *Transactions of the Society of Computer Simulation International*, 13(2):87–101, 1996.

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

[33] M. Tiller. *Introduction to Physical Modeling With Modelica*, volume 615 of *International Series in Engineering and Computer Science*. Kluwer, 2001.

[34] L. Todorovski and S. Džeroski. Using domain knowledge on population dynamics modeling for equation discovery. In *Proceedings of the 12th European Conference on Machine Learning*, pages 478–490, 2001.

[35] A. van Ackere and P. Smith. Towards a macro model of national health service waiting lists. *System Dynamics Review*, 15(3):225–252, 1999.

[36] P. Verhulst. 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, 1838.

[37] D. Waltz. Understanding line drawings of scenes with shadows. In Winston, P., editor, *The Psychology of Computer Vision*, pages 19–91. McGraw-Hill, 1975.

Appendix

Theorem 1 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 δ.
Proof: Let A be the smallest set of assumptions such that $A, \Delta \vdash M$, A_δ denote the instantiation of the assumptions of the disaggregation fragment and $\neg A_\delta$ denote the set of assumptions corresponding to the negation of the assumptions in A_δ. The only change ApplyDF makes to the original model space consists of extending the justifications of certain nodes with $\neg A_\delta$. As the assumption specifications in A_δ do not refer to assumption instances that are referred to by the model fragments in the knowledge base, and hence in Δ (definition 4), $A \cup \neg A_\delta \nvdash \bot$. Therefore, a set of assumptions $A' \subseteq A \cup A_\delta$ exists such that $A', \Delta' \vdash M$. ■

Theorem 2 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 δ to a set of model fragment nodes M, (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 derived by $A \cup A_\delta, \Delta' \vdash M_D$ is a disaggregate model of M_A
Proof: For a given set of assumptions X, let $P(X)$ denote the set of participants $\{p \mid X, \Delta' \vdash p\}$, and $R(X)$ denote the set of relations $\{r \mid X, \Delta' \vdash r\}$. Thus, $M_D = \langle P(A \cup A_\delta), R(A \cup A_\delta)\rangle$. It follows from theorem 1 that each aggregate model M_A that follows from the set of assumptions A in model space Δ also follows from $A \cup \neg A_\delta$, and hence $M_A = \langle P(A \cup \neg A_\delta), R(A \cup \neg A_\delta)\rangle$.

As $P(A) \subseteq P(A \cup B)$ and $R(A) \subseteq R(A \cup B)$, $P(A)$ (or $R(A)$) is a set of participants (or relations) that M_A and M_D have in common. The nodes in $P(A \cup A_\delta) - P(A)$ (or $R(A \cup A_\delta) - R(A)$) are derived from the nodes that depend on the model fragment instances to which the disaggregation fragment is applied. They can be of two types:

- Some nodes represent participants or relations that are disaggregated according to the disaggregation fragment. For each of the participants $p \in P_{\text{dom}(g_P)}$ (with $P_{\text{dom}(g_P)} = P(A \cup A_\delta) - P(A) \cap \text{dom}(g_P)$), the set of participants $P_{\text{range}(g_P)} = \{g_P(\ldots, n_i, \ldots, p) \mid n_i \in N_i\}$ is created. The other participants (denoted P_a) are copied into new nodes whose label is the combination of the original label and A. Similarly, for each of the relations $r \in R_{\text{dom}(g_R)}$ (with $R_{\text{dom}(g_R)} = R(A \cup A_\delta) - R(A) \cap \text{dom}(g_R)$), a new set of relations $R_{\text{range}(g_R)} = \{g_R(\ldots, n_i, \ldots, r) \mid n_i \in N_i\}$ is created. Following definition 4, based on the remaining relations in $r(p^a_q, \ldots, p^a_r) \in R_a$ (with $R_a = R(A \cup \neg A_\delta) - \text{dom}(g_R)$), the set $R_d = \{r(g'_P(\ldots, n_i, \ldots, p^a_q), \ldots, g'_P(\ldots, n_i, \ldots, p^a_r)) \mid n_i \in N_i\}$ is created.

- The remaining nodes are the participants and relations that are newly instantiated from the target-participants and postconditions respectively. They are denoted by the sets P' and R'.

From this, it follows that $M_A = \langle P(A) \cup P_a \cup P_{\text{dom}(g_P)}, R(A) \cup R_a \cup R_{\text{dom}(g_R)} \rangle$ and that $M_D = \langle P(A) \cup P_a \cup P_{\text{range}(g_P)} \cup P', R(A) \cup R_{\text{range}(g_R)} \cup R_d \cup R' \rangle$. For the substitutions $P^c = P(A) \cup P^a$, $P^a = P_{\text{dom}(g_P)}$, $P^d = R_{\text{dom}(g_R)}$, $R = R(A)$, $R^a = R_a \cup R_{\text{dom}(g_R)}$ and $R^d =_{\text{range}(g_R)} \cup R_d$, definition 2 applies. ■

Theorem 3 The combined disaggregation of the model space is a commutative operation. In other words,

$$D_\theta(\Delta, \delta_2 \circ \delta_1, M_1 \cup M_2 \cup M_{12}) = D_\theta(\Delta, \delta_1 \circ \delta_2, M_1 \cup M_2 \cup M_{12})$$

Proof: Following definition 7, this theorem needs to prove that, given two disaggregation fragments δ_1 and δ_2, a model space Δ and a partition $\{M_{12}, M_1, M_2, M\}$ of those nodes in Δ which represent model fragment instances, where $M_{12} \cup M_1$ and $M_{12} \cup M_2$ are minimal sets for model fragments to which δ_1 and δ_2 can be respectively applied,

$$\begin{aligned}
&d_\theta(d_\theta(\Delta, \delta_1, M_1 \cup M_{12}), \delta_2, d_M(\Delta, \delta_1, M_{12}) \cup M_2) \cup \\
&d_\theta(d_\theta(\Delta, \delta_1, M_1 \cup M_{12}), \delta_1, P^t(\delta_2, M_2 \cup M_{12})) \cup \\
&d_\theta(d_\theta(\Delta, \delta_1, M_1 \cup M_{12}), \delta_1, \Phi^t(\delta_2, M_2 \cup M_{12})) = \\
&d_\theta(d_\theta(\Delta, \delta_2, M_2 \cup M_{12}), \delta_1, d_M(\Delta, \delta_2, M_{12}) \cup M_1) \cup \\
&d_\theta(d_\theta(\Delta, \delta_2, M_2 \cup M_{12}), \delta_2, P^t(\delta_1, M_1 \cup M_{12})) \cup \\
&d_\theta(d_\theta(\Delta, \delta_2, M_2 \cup M_{12}), \delta_2, \Phi^t(\delta_1, M_1 \cup M_{12}))
\end{aligned}$$

Based on these notations, the following properties can be established:

Property 1. For each pair of sets of model fragment instances M_a and M_b, such that $M_a \cap M_b = \emptyset$,

$$d_\theta(\Delta, d, M_a \cup M_b) = d_\theta(\Delta, d, M_a) \cup d_\theta(\Delta, d, M_b)$$

When compared to Δ, $d_\theta(\Delta, d, M)$ consists of a new sub-hypergraph for each model fragment instance $m \in M$, containing the disaggregations (according to d) of the consequents of m. It is equivalent whether to apply this operation to the set M as a whole, or to apply it to the sets of a partition of M is equivalent.

Property 2. For each pair of disaggregation fragments δ_a and δ_b and each pair of sets of model fragments M_a and M_b, such that $M_a \cap M_b = \emptyset$, to which δ_a and δ_b are applicable,

$$\begin{aligned}
&d_\theta(d_\theta(\Delta, \delta_b, M_b), \delta_a, d_M(\Delta, \delta_b, M_a)) \\
=&d_\theta(d_\theta(\Delta, \delta_a, M_a), \delta_b, d_M(\Delta, \delta_a, M_b)) \\
=&d_\theta(\Delta, \delta_a, M_a) \cup d_\theta(\Delta, \delta_b, M_b)
\end{aligned}$$

As δ_a (or δ_b) is applied to a set of model fragments M_a (or M_b) that does not intersect with the model fragments to which the other disaggregation fragment δ_b (or δ_a) is applied, $d_M(\Delta, \delta_a, M_b) = M_b$ (or $d_M(\Delta, \delta_b, M_a) = M_a$). Thus, the order in which the disaggregation fragments are applied is irrelevant.

Property 3. For each pair of disaggregation fragments δ_a and δ_b that are applied to the same set of model fragments M in a model space Δ,

$$\begin{aligned}
&d_\theta(d_\theta(\Delta, \delta_b, M), \delta_a, d_M(\Delta, \delta_b, M)) \cup d_\theta(\Delta, \delta_a, M) \\
=&d_\theta(d_\theta(\Delta, \delta_a, M), \delta_b, d_M(\Delta, \delta_a, M)) \cup d_\theta(\Delta, \delta_b, M)
\end{aligned}$$

The application of the first disaggregation fragment, say $d_\theta(\Delta, \delta_a, M)$, creates a new set of sub-hypergraphs with roots $d_M(\Delta, \delta_a, M)$. The second disaggregation fragment will copy these into a new set of sub-hypergraphs with roots $d_M(d_\theta(\Delta, \delta_a, M), \delta_b, d_M(\Delta, \delta_a, M))$ in which the consequents are disaggregated according to δ_a and δ_b. The reverse order of application of disaggregation fragments also produces a new sub-hypergraphs containing the consequents of M disaggregated according to δ_a and δ_b, but through the different intermediate $d_\theta(\Delta, \delta_b, M)$. Therefore

$$\begin{aligned}
&d_\theta(d_\theta(\Delta, \delta_b, M), \delta_a, d_M(\Delta, \delta_b, M)) - d_\theta(\Delta, \delta_b, M) \\
=&d_\theta(d_\theta(\Delta, \delta_a, M), \delta_b, d_M(\Delta, \delta_a, M)) - d_\theta(\Delta, \delta_a, M)
\end{aligned}$$

Theorem 4 Given two disaggregation fragments δ_1 and δ_2, a model space Δ and a partition $\{M_{12}, M_1, M_2, M\}$ of those nodes in Δ which represent model fragment instances, where $M_{12} \cup M_1$ and $M_{12} \cup M_2$ are minimal sets for model fragments to which δ_1 and δ_2 can be respectively applied,

$$\begin{aligned} & d_\theta(d_\theta(\Delta, \delta_1, M_{12} \cup M_1), \delta_2, d_M(\Delta, \delta_1, M_{12} \cup M_2)) \\ =& d_\theta(d_\theta(\Delta, \delta_2, M_{12} \cup M_2), \delta_1, d_M(\Delta, \delta_2, M_{12} \cup M_1)) \end{aligned}$$

Based on the previous properties:

$$\begin{aligned} & d_\theta(d_\theta(\Delta, \delta_1, M_{12} \cup M_1), \delta_2, d_M(\Delta, \delta_1, M_{12} \cup M_2)) \\ = \; & \left.\begin{array}{l} d_\theta(d_\theta(\Delta, \delta_1, M_{12}), \delta_2, d_M(\Delta, \delta_1, M_{12})) \cup \\ d_\theta(d_\theta(\Delta, \delta_1, M_1), \delta_2, d_M(\Delta, \delta_1, M_{12})) \cup \\ d_\theta(d_\theta(\Delta, \delta_1, M_{12}), \delta_2, M_2) \cup \\ d_\theta(d_\theta(\Delta, \delta_1, M_1), \delta_2, M_2) \end{array}\right\} \quad \text{(due to property 1)} \\ = \; & \left.\begin{array}{l} d_\theta(d_\theta(\Delta, \delta_1, M_{12}), \delta_2, d_M(\Delta, \delta_1, M_{12})) \cup \\ d_\theta(\Delta, \delta_1, M_1) \cup d_\theta(\Delta, \delta_1, M_{12}) \cup \\ d_\theta(\Delta, \delta_2, M_2) \cup d_\theta(\Delta, \delta_2, M_{12}) \end{array}\right\} \quad \text{(due to property 2)} \end{aligned} \tag{21}$$

Following the same line for reasoning, it can be proven that:

$$\begin{aligned} & d_\theta(d_\theta(\Delta, \delta_2, M_{12} \cup M_2), \delta_1, d_M(\Delta, \delta_2, M_{12} \cup M_1)) \\ = \; & d_\theta(d_\theta(\Delta, \delta_2, M_{12}), \delta_1, d_M(\Delta, \delta_2, M_{12})) \cup \\ & d_\theta(\Delta, \delta_1, M_1) \cup d_\theta(\Delta, \delta_1, M_{12}) \cup \\ & d_\theta(\Delta, \delta_2, M_2) \cup d_\theta(\Delta, \delta_2, M_{12}) \end{aligned}$$

From property 3

$$\begin{aligned} & d_\theta(d_\theta(\Delta, \delta_1, M_{12}), \delta_2, d_M(\Delta, \delta_1, M_{12})) \cup d_\theta(\Delta, \delta_2, M_{12}) \\ =& d_\theta(d_\theta(\Delta, \delta_2, M_{12}), \delta_1, d_M(\Delta, \delta_2, M_{12})) \cup d_\theta(\Delta, \delta_1, M_{12}) \end{aligned}$$

it follows that:

$$\begin{aligned} & d_\theta(d_\theta(\Delta, \delta_1, M_{12} \cup M_1), \delta_2, d_M(\Delta, \delta_1, M_{12} \cup M_2)) \\ =& d_\theta(d_\theta(\Delta, \delta_2, M_{12} \cup M_2), \delta_1, d_M(\Delta, \delta_2, M_{12} \cup M_1)) \end{aligned}$$

which proves this sub-theorem.

Property 4. For each pair of model spaces (or parts of model spaces) Δ_a and Δ_b, such that $\Delta_a \cap \Delta_b = \emptyset$,

$$d_\theta(\Delta_a \cup \Delta_b, d, M) = d_\theta(\Delta_a, d, M) \cup d_\theta(\Delta_b, d, M) \tag{22}$$

Now, the required proof can proceed. From definition 6 and property 4:

$$\begin{aligned} & D_\theta(D_\theta(\Delta, \delta_1, M_{12} \cup M_1), \delta_2, d_M(\Delta, \delta_1, M_{12} \cup M_2)) \cup \\ & d_\theta(P^t(\Delta, \delta_2, M_{12} \cup M_1), \delta_1, M_{12} \cup M_1 \cup M_2) \cup \\ & d_\theta(\Phi^t(\Delta, \delta_2, M_{12} \cup M_1), \delta_1, M_{12} \cup M_1 \cup M_2) \end{aligned} \tag{23}$$

$$\begin{aligned} =& D_\theta(d_\theta(\Delta, \delta_1, M_{12} \cup M_1) \cup \\ & P^t(d_\theta, \delta_1, M_{12} \cup M_1) \cup \Phi^t(d_\theta, \delta_1, M_{12} \cup M_1), \delta_2, d_M(\Delta, \delta_1, M_{12} \cup M_2)) \cup \\ & d_\theta(P^t(\Delta, \delta_2, M_{12} \cup M_1), \delta_1, M_{12} \cup M_1 \cup M_2) \cup \\ & d_\theta(\Phi^t(\Delta, \delta_2, M_{12} \cup M_1), \delta_1, M_{12} \cup M_1 \cup M_2) \\ =& D_\theta(d_\theta(\Delta, \delta_1, M_{12} \cup M_1), \delta_2, d_M(\Delta, \delta_1, M_{12} \cup M_2)) \cup \\ & D_\theta(P^t(d_\theta, \delta_1, M_{12} \cup M_1), \delta_2, d_M(\Delta, \delta_1, M_{12} \cup M_2)) \cup \\ & D_\theta(\Phi^t(d_\theta, \delta_1, M_{12} \cup M_1), \delta_2, d_M(\Delta, \delta_1, M_{12} \cup M_2)) \cup \\ & d_\theta(P^t(\Delta, \delta_2, M_{12} \cup M_1), \delta_1, M_{12} \cup M_1 \cup M_2) \cup \\ & d_\theta(\Phi^t(\Delta, \delta_2, M_{12} \cup M_1), \delta_1, M_{12} \cup M_1 \cup M_2) \end{aligned} \tag{24}$$

For the same reasons, it can be proven that:

$$
\begin{aligned}
& D_\theta(D_\theta(\Delta, \delta_2, M_{12} \cup M_2), \delta_1, d_M(\Delta, \delta_2, M_{12} \cup M_1)) \cup \\
& d_\theta(P^t(\Delta, \delta_1, M_{12} \cup M_2), \delta_2, M_{12} \cup M_2 \cup M_1) \cup \\
& d_\theta(\Phi^t(\Delta, \delta_1, M_{12} \cup M_2), \delta_2, M_{12} \cup M_2 \cup M_1) \qquad (25) \\
= & D_\theta(d_\theta(\Delta, \delta_2, M_{12} \cup M_2), \delta_2, d_M(\Delta, \delta_2, M_{12} \cup M_1)) \cup \\
& D_\theta(P^t(d_\theta, \delta_2, M_{12} \cup M_2), \delta_1, d_M(\Delta, \delta_2, M_{12} \cup M_1)) \cup \\
& D_\theta(\Phi^t(d_\theta, \delta_2, M_{12} \cup M_2), \delta_1, d_M(\Delta, \delta_2, M_{12} \cup M_1)) \cup \\
& d_\theta(P^t(\Delta, \delta_1, M_{12} \cup M_2), \delta_2, M_{12} \cup M_2 \cup M_1) \cup \\
& d_\theta(\Phi^t(\Delta, \delta_1, M_{12} \cup M_2), \delta_2, M_{12} \cup M_2 \cup M_1) \qquad (26)
\end{aligned}
$$

Because of theorem 4, (24) is equivalent to (26). It follows that (23) is equivalent to (25). This proves that commutativity holds for the combined applications of disaggregation fragments. ■