

Optimizing Distributed Resource Allocation using Epistemic Game Theory: A Model-driven Engineering Approach

Fazle Rabbi^{1,2}, Lars Michael Kristensen¹ and Yngve Lamo¹

¹Bergen University College, Bergen, Norway

²University of Oslo, Oslo, Norway

Keywords: Metamodelling, Epistemic Game Theory, Model Transformation, Optimization, Distributed Systems.

Abstract: Distributed systems modelling often involves a set of heterogeneous models where each model specifies a set of local constraints capturing a specific view of the system. In real life, distributed systems are often loosely connected and interdependencies are not defined into their software models. This limits the scope of optimization of distributed resources. In this paper, we merge heterogeneous models of distributed systems and articulate distributed resource constraints via inter-metamodel constraints. We apply model-driven engineering and use model transformation rules to construct an epistemic game theory model for the purpose of optimizing distributed resource allocation. Since the application of transformation rules normally do not guarantee the satisfaction of constraints when applied on a model, it requires a conformance checking which is an expensive operation. To overcome this problem, we introduce the concept of *compliant rule* for efficient model transformation.

1 INTRODUCTION

Distributed systems are organized in a modular, loosely coupled fashion such as in healthcare organizations, where separate departments deliver different types of healthcare. Each unit in a healthcare organization deals with a different set of operational constraints (Pinelle and Gutwin, 2006). Patients often require a diverse set of services which are provided by different health facilities. However, the medical care provided by various health facilities are not necessarily optimized to reduce patients' waiting time and other important aspects of getting healthcare services.

While optimizing the resource allocation of a healthcare system, game theory is more suitable choice as there are many participants and it is required to reason about optimality respecting patients preferences and priorities. Healthcare systems need to be equipped with uncertainty management as many patients attending do not have an obvious diagnosis at presentation; also there are varieties of uncertainty in healthcare (Han et al., 2011). We propose to use epistemic game theory for optimizing the use of healthcare resources with the aim of improving the quality and efficiency of healthcare services. Game theory is the discipline of science that models strategic situations where individuals reason about others choices

for decision making. Epistemic game theory (Perea, 2012; Perea, 2014) can be viewed as rational decision making under uncertainty. It can play an important role in coordinating distributed systems or in the design of interconnected systems. A model driven approach for meta-modelling epistemic game theory was introduced in (Rabbi et al., 2016a). We investigated how distributed systems may be coordinated to optimize their resources respecting a distributed set of *local constraints* (Rabbi et al., 2016a). However, there are situations where loosely coupled distributed systems have to deal with a set of *global constraints*. For instance, patients scheduled appointment time for getting a healthcare service should not conflict with other scheduled appointments for getting other healthcare services. In this paper, we enhance the approach with inter-metamodel (i.e., global) constraints.

We present our approach with a running example from the healthcare domain. Let us consider that you and Barbara are two orthopedic patients at the same hospital. Both you and Barbara are asked to visit the department of orthopedic surgery on the same day at 9am. When you report to the front desk of the orthopedic department you are asked to take an X-ray from the radiology department which is located on the same floor as the orthopedic department. The ortho-

pedic department sends an imaging order electronically to the radiology department which includes patients information and an instruction about the X-ray. It took some time for you to reach to the radiology department as you have a long leg cast. Barbara reported to the front desk right after you, but she reached the radiology department before you. The radiology department is equipped with a queue automation and therefore you will be served after Barbara. While you are waiting for your turn at the radiology department you do not know that Dr. Logan (your doctor) will be out soon for his round at the inpatient medical wards. Barbara also does not know that she will not be served at the orthopedic department before 10.30am because there are more patients waiting to see her doctor Dr. Bryan. When you are done with the X-ray and return to the orthopedic department, you find that Dr. Logan is out for the rounds and you will have to wait until 12pm. It could have been an optimal decision to have your X-ray done before Barbara as Barbara will have to wait until 10.30am anyway. Even though the orthopedic department and the radiology department are located on the same floor they do not share their appointment schedules and as a result, the overall wait time for patients to get health services are increased. We propose to use epistemic game theory for optimizing the use of distributed resources and add resource constraints to game theory. This paper presents a formal treatment of distributed resource constraints via inter-metamodel constraints. For the efficient construction of game theory models we introduce the idea of compliant rules.

The paper is organized as follows: In section 2, we present a formal model for distributed resource constraints. Section 3 presents a model transformation technique for resource allocation. Section 4 presents a game theory model for optimizing resource allocation. Section 5 introduces compliant rules and coordination of transformation rules, section 6 presents some related works and section 7 concludes the paper.

2 MODELLING RESOURCE CONSTRAINTS

Our approach is based on category theory (Barr and Wells, 1995), model transformations (Ehrig et al., 2006) and epistemic game theory (Perea, 2012). We use Diagrammatic Logic (Diskin and Wolter, 2008) and the Diagram Predicate Framework (DPF) (Rutle, 2010) for the formal development of metamodel specifications. DPF is suitable for developing domain specific modelling languages and has support for an unbounded number of metalevels. In DPF, a model

is represented by a diagrammatic specification $\mathfrak{S} = (S, C^{\mathfrak{S}} : \Sigma)$ consisting of an underlying graph S together with a set of *atomic constraints* $C^{\mathfrak{S}}$ specified by a *predicate signature* Σ . A predicate signature consists of a collection of predicates, each having a name, an arity (or shape graph), a visualization, and a semantic interpretation (see Table 1). A predicate is used to specify a constraint in a model by means of graph homomorphisms. An atomic constraint (p, δ) is added to a graph S by a predicate symbol p and a graph homomorphism $\delta : \alpha^{\Sigma}(p) \rightarrow S$ where $\alpha^{\Sigma}(p)$ represents the shape graph of predicate p . DPF provides a formalization of multi level meta-modelling by defining the conformance relation between models at adjacent levels of a meta-modelling hierarchy. There are two kinds of conformance: *typed by* and *satisfaction of constraints*. Figure 1(i) and (ii) shows the DPF metamodel specifications of an orthopedic department (\mathfrak{S}^1) and a radiology department (\mathfrak{S}^2), respectively. These specifications are constrained by predicates from Table 1.

Constraints are added into the specifications by graph homomorphisms from the shape graph of the predicates to the model elements. Below is a list of constraints specified in \mathfrak{S}^1 and \mathfrak{S}^2 :

- **C1.** *A patient must have exactly one birthdate in an instance of \mathfrak{S}^1 (specified by $\langle \text{mult}(1,1) \rangle$)*
- **C2.** *A person must have exactly one birthdate in an instance of \mathfrak{S}^2 (specified by $\langle \text{mult}(1,1) \rangle$)*
- **C3.** *An appointment time-slot (i.e., $TS@Dept$) allocated to a patient in an instance of \mathfrak{S}^1 must belong to that patients assigned doctor (specified by $\langle \text{composite} \rangle$)*
- **C4.** *A person can only check-in for an examination in an instance of \mathfrak{S}^2 if the person has an imaging order (specified by $\langle \text{pre-condition} \rangle$)*
- **C5.** *Only registered persons are allocated with examination time-slots (i.e., $TS@Lab$) in an instance of \mathfrak{S}^2 (specified by $\langle \text{pre-condition} \rangle$)*
- **C6.** *An appointment time-slot in an instance of \mathfrak{S}^1 must not be allocated to more than one patient (specified by $\langle \text{injective} \rangle$)*
- **C7.** *An exam time-slot in an instance of \mathfrak{S}^2 must not be allocated to more than one person (specified by $\langle \text{injective} \rangle$)*

2.1 Distributed Resource Constraints

Usually orthopedic doctors need to inspect the X-ray reports of the patient during consultation. Therefore

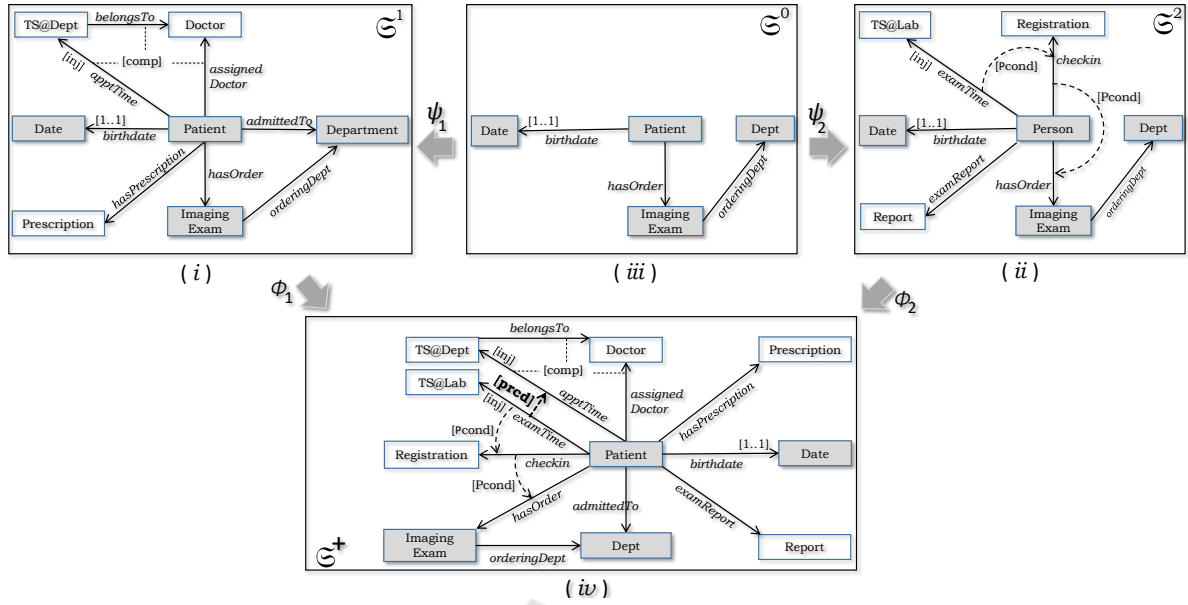


Figure 1: (i) Metamodel specification of an orthopedic department, (ii) Metamodel specification of a radiology department, (iii) overlap specification, (iv) merged metamodel specification with an inter-metamodel constraint.

Table 1: Predicates of a signature, Σ .

p	Arity $\alpha_{\Sigma}^{\sum}(p)$	Visualization	Semantic interpretation
$\langle \text{mult}(n,m) \rangle$	$1 \xrightarrow{f} 2$	$X \xrightarrow{f} [n..m] Y$	f must have at least n and at most m instances for each instance of X
$\langle \text{pre-condition} \rangle$	$1 \xrightarrow{f} 2$ $1 \xrightarrow{g} 3$	$X \xrightarrow{f} Y$ $X \xrightarrow{g} Z$ [Pcond]	For each instance of f there exists an instance of g with the same source node
$\langle \text{composite} \rangle$	$1 \xrightarrow{f} 2$ $1 \xrightarrow{g} 3$ $1 \xrightarrow{h} 3$	$X \xrightarrow{f} Y$ $X \xrightarrow{g} Z$ $X \xrightarrow{h} Z$ [comp]	For each composition of instances f, g , there exists an instance of h
$\langle \text{preceed} \rangle$	$1 \xrightarrow{f} 2$ $1 \xrightarrow{g} 3$	$X \xrightarrow{f} Y$ $X \xrightarrow{g} Z$ [precd]	If there are instances of f and g with the same source node, then there exists an ordering of values between instances of Y and Z
$\langle \text{injective} \rangle$	$1 \xrightarrow{f} 2$	$X \xrightarrow{f} [inj] Y$	Instances of f never maps distinct elements of its domain to the same element of its codomain

orthopedic patients' time-slot at the radiology department must be preceded by the time-slot at the orthopedic department. We will refer to this constraint as **C8**. We use the concept 'time-slot' for an appointment or an examination to represent the time assigned for a scheduled job. A time-slot $t2$ which include information about the start-time and end-time of a scheduled job is preceded by a time-slot $t1$ if the end-time of $t2$ are less than the end-time of $t1$. To model this distributed resource constraint, we need to merge specifications \mathfrak{S}^1 and \mathfrak{S}^2 .

While merging specifications, atomic constraints specified in the component metamodel specifications need to be preserved. We merge specifications in such a way that for any component metamodel specification \mathfrak{S}^i there exists a specification morphism from \mathfrak{S}^i to the merged metamodel specification. The definition of specification morphism is given below (Rutle, 2010):

Definition 1 (Specification Morphism). *Given two specifications $\mathfrak{S} = (S, C^{\mathfrak{S}} : \Sigma)$ and $\mathfrak{S}' = (S', C^{\mathfrak{S}'} : \Sigma)$, a specification morphism $\phi : \mathfrak{S} \rightarrow \mathfrak{S}'$ is a graph homomorphism $\phi : S \rightarrow S'$ such that for any atomic constraint $(p, \delta) \in C^{\mathfrak{S}}$ implies $(p, \delta; \phi) \in C^{\mathfrak{S}'}$ as illustrated in the following diagram.*

$$\alpha_{\Sigma}^{\sum}(p) \xrightarrow{\delta} S \xrightarrow{\phi} S'$$

$\delta; \phi$

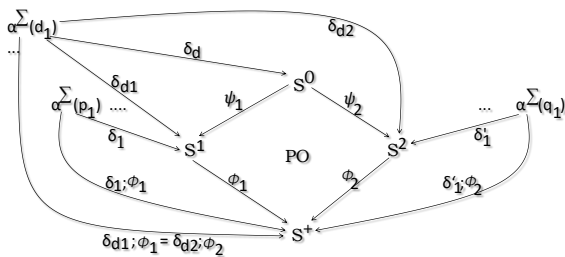
Diskin et al. presented an approach of merging metamodels by making metamodels explicit and by introducing a correspondence span between component metamodels (Diskin et al., 2010) (Diskin, 2011). We adapt the approach with specification morphisms in order to preserve constraints into the merged metamodel specifications. Details of merging heterogeneous metamodels and specifying inter-metamodel constraints can be found in (Diskin et al., 2010). Unlike (Diskin et al., 2010), in our approach the head of the span is an *overlap specification* given in DPF which specifies the common concepts of both metamodels and the legs of the span are specification

morphisms. Figure 1(iii) shows the overlap specification $\mathfrak{E}^0 = (S^0, C^{\mathfrak{E}^0} : \Sigma)$ (highlighted in grey) and Figure 1(iv) shows the merged metamodel specification $\mathfrak{E}^+ = (S^+, C^{\mathfrak{E}^+} : \Sigma)$. \mathfrak{E}^+ is constructed by merging component metamodel specifications \mathfrak{E}^1 and \mathfrak{E}^2 modulo their overlap \mathfrak{E}^0 (i.e., common elements appear only once in the merge). Overlaps are specified by two specification morphisms $\mathfrak{E}^1 \xleftarrow{\psi_1} \mathfrak{E}^0 \xrightarrow{\psi_2} \mathfrak{E}^2$ where \mathfrak{E}^0 contains all common concepts, S^1 and S^2 are the underlying graphs of specifications \mathfrak{E}^1 and \mathfrak{E}^2 . Any pair of elements $x_1 \in S^1$ and $x_2 \in S^2$ are considered to be the same if there is an element $x \in S^0$ such that $\psi_1(x) = x_1$ and $\psi_2(x) = x_2$. Following (Diskin et al., 2010), we call this configuration of metamodel specifications and mappings a multi-metamodel specification \mathcal{M} , and write $\mathcal{M} = (\mathfrak{E}^1, \mathfrak{E}^2, \mathfrak{E}^0, \psi_1, \psi_2)$. The multi-metamodel specification \mathcal{M} declares the unification of concepts *Date*, *birthdate*, *hasOrder*, *ImagingExam*, *orderingDept* and unifies also the following concepts:

- *Patient* and *Person* since $\psi_1(\text{Patient}) = \text{Patient}$ and $\psi_2(\text{Patient}) = \text{Person}$
- *Department* and *Dept* since $\psi_1(\text{Dept}) = \text{Department}$ and $\psi_2(\text{Dept}) = \text{Dept}$

The mappings $\phi_1 : \mathfrak{E}^1 \rightarrow \mathfrak{E}^+$ and $\phi_2 : \mathfrak{E}^2 \rightarrow \mathfrak{E}^+$ in Figure 1 maps elements from \mathfrak{E}^1 and \mathfrak{E}^2 to the corresponding elements in \mathfrak{E}^+ .

Definition 2 (Merged Metamodel Specification). *Given two (component) metamodel specifications $\mathfrak{E}^1 = (S^1, C^{\mathfrak{E}^1} : \Sigma)$ and $\mathfrak{E}^2 = (S^2, C^{\mathfrak{E}^2} : \Sigma)$, a merged metamodel specification $\mathfrak{E}^+ = (S^+, C^{\mathfrak{E}^+} : \Sigma)$ of \mathfrak{E}^1 and \mathfrak{E}^2 is given by two specification morphisms $\phi_1 : \mathfrak{E}^1 \rightarrow \mathfrak{E}^+$, $\phi_2 : \mathfrak{E}^2 \rightarrow \mathfrak{E}^+$ and a multi-metamodel specification $\mathcal{M} = (\mathfrak{E}^1, \mathfrak{E}^2, \mathfrak{E}^0, \psi_1, \psi_2)$ such that the graph S^+ is obtained by the pushout of ψ_1 and ψ_2 . This is illustrated in the following diagram:*



The binary case can be generalized to the N-ary case by constructing S^+ as a so-called colimit and preserving all the atomic constraints specified in the component metamodels. Constraints of component metamodel specifications are preserved in a merged

metamodel specification. Additional constraints may be added into a merged metamodel specification.

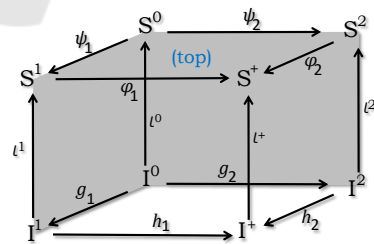
The distributed resource constraints are added into the merged metamodel specification by means of inter-metamodel constraints. The distributed resource constraint **C8** is added into \mathfrak{E}^+ by using the predicate $\langle precede \rangle$. In general, merged metamodel specifications have the following properties:

Theorem 1. *Let \mathfrak{E}^+ be a merged metamodel specification obtained from component metamodel specifications $\mathfrak{E}^1, \dots, \mathfrak{E}^n$. For any component metamodel specification $\mathfrak{E}^i (1 \leq i \leq n)$, there exists a specification morphism from \mathfrak{E}^i to \mathfrak{E}^+ .*

Proof. This follows directly from the definition of merged metamodel specification. \square

A model (I, ι) of a metamodel specification $\mathfrak{E} = (S, C^{\mathfrak{E}} : \Sigma)$ is a graph I together with a typing graph homomorphism $\iota : I \rightarrow S$. In order to be a valid model of a metamodel specification \mathfrak{E} the typing morphism must satisfy the atomic constraints specified in $C^{\mathfrak{E}}$. Here we define the merged model of two component models.

Definition 3. *Given two (component) models $(I^1, \iota^1), (I^2, \iota^2)$ of metamodel specifications $\mathfrak{E}^1, \mathfrak{E}^2$ and an overlap model $I^0 \xleftarrow{g_1} I^1 \xrightarrow{g_2} I^2$ typed by the overlap specification of \mathfrak{E}^1 and \mathfrak{E}^2 . The merged model is a graph I^+ together with a typing graph homomorphism $\iota^+ : I^+ \rightarrow S^+$ where S^+ is the underlying graph of the merged metamodel specifications of $\mathfrak{E}^1, \mathfrak{E}^2$ such that the graph I^+ is obtained by the pushout of g_1 and g_2 and the top face constitute a van Kampen square in the category of Graph as illustrated in the following diagram:*



The reader interested in the property of van Kampen square may wish to consult (Mantz, 2014). It is possible that the atomic constraints when preserved in a merged metamodel specification may conflict with other constraints or may become redundant. There exists some research for detecting these problems automatically such as the alloy analyzer which may be used to detect these problems by specification analysis (Wang et al., 2015). An atomic constraint specified in a component metamodel and satisfied in a com-

ponent model may become unsatisfiable in a merged model or vice versa. Figure 2 shows an example of a model merge where a component model (I^1, ι^1) is typed by the underlying graph S^1 of the metamodel specification \mathfrak{S}^1 . The predicate p is used to constraint \mathfrak{S}^1 by a graph homomorphism $\delta: \alpha^\Sigma(p) \rightarrow S^1$.

The edge $1 \xrightarrow{e_1} 2$ is mapped to $A \xrightarrow{f} B$ and the edge $2 \xrightarrow{e_2} 1$ is mapped to $B \xrightarrow{g} A$. The overlap specification is given by $(\mathfrak{S}^1, \mathfrak{S}^2, \mathfrak{S}^0, \psi_1, \psi_2)$ which unifies of the following concepts: A and AA , B and BB . To check if (I^1, ι^1) satisfies the constraint (p, δ) we extract the fragment (O^*) of the component model that is affected by the constraint. Again, to check if the merged model (I^+, ι_m) satisfies the constraint $(p, \delta; \phi_1)$ we extract the fragment (O_m^*) of the merged model that is affected by the constraint. The object O^* is obtained by the pullback of $\alpha^\Sigma(p) \xrightarrow{\delta} S^1 \xleftarrow{\iota^1} I^1$ and the object O_m^* is obtained by the pullback of $\alpha^\Sigma(p) \xrightarrow{\delta; \phi_1} S^+ \xleftarrow{\iota_m} I^+$. Since we obtain two different objects by the pullback operation, it is possible that the satisfaction of the local constraint will vary.

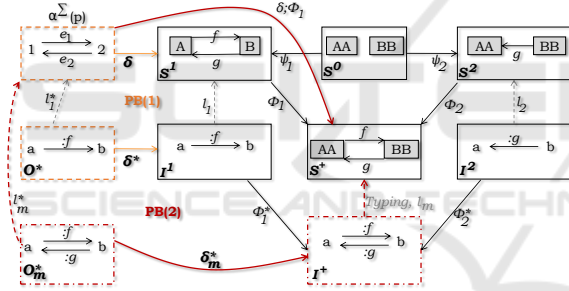


Figure 2: Construction of pullback for a component model I^1 and merged model I^+ .

3 RESOURCE ALLOCATION

Our example introduced earlier consists of two component metamodels representing two distributed systems and an inter-metamodel constraint **C8**. Figure 3 shows a model (I, ι_I) of the merged metamodel \mathfrak{S}^+ of Figure 1. The typing ι_I of the modelling elements are not explicitly represented in the figure. In the figure, *You* and *Barbara* are instances of *Patient*, your assigned doctor is *Dr. Logan* and Barbara's assigned doctor is *Dr. Bryan*. Dr. Logan has two available time-slots: 0950 – 1010@*Logan* and 1200 – 1220@*Logan* and Dr. Bryan has one available time-slot: 1030 – 1050@*Bryan*. There are two available time-slots at the radiology department: 0945 – 1000@*Lab* and 0930 – 0945@*Lab*. In time-slot 0950 – 1010@*Logan*, 09:50 is the start time and 10:10 is the end time.

Model (I, ι_I) need to be completed with resource allocation for you and Barbara at the radiology and orthopedic department. One can find that there are four possible choices of resource allocation for you and two possible choices of resource allocation for Barbara with the available resources.

We use a transformation approach where transformation rules have a set of negative application conditions proposed by Lambers et al., in (Lambers et al., 2008) for allocating resources to the patients. Figure 4 illustrates a rule r_1 typed by the merged metamodel \mathfrak{S}^+ . The typing information of a modelling element in r_1 appears after a colon (:). The rule specifies that a patient pt_1 is allocated with an appointment time-slot $t1$ and an exam time-slot $t2$ if $t1$ belongs to pt_1 's assigned doctor d and $t1, t2$ are not allocated to any other patient. The green color is used to indicate elements that the rule is going to produce. Negative application conditions (NACs) are typically used in graph transformation to prohibit an infinite number of rule applications.

The rule can be applied over the model (I, ι_I) in six different ways since there are six matches. Figure 3 shows briefly the choices of resource allocation for you in $y_1 - y_4$ and for Barbara in b_1, b_2 . y_1 shows a resource allocation where you are assigned with exam time-slot 0930 – 0945@*Lab* at the radiology department and appointment time-slot 0950 – 1010@*Logan* at the orthopedics department with Dr. Logan. This resource distribution is valid since it is not violating any constraint. Similarly, other valid resource distributions are y_2, y_3, b_1 and b_2 . However y_4 is not valid as it is violating the distributed resource constraint **C8**: your exam time-slot 0945 – 1000@*Lab* must be preceded by your appointment time-slot 0950 – 1010@*Logan*.

We apply epistemic game theory for the optimization of resources over the valid choices of resource allocation. Epistemic game theory is a broad area of research that formalizes the assumptions about rationality and mutual beliefs in a formal language to analyze games. It introduces a Bayesian perspective on decision-making in strategic situations. In a strategic game, the outcome does not only depend on the choice of a single player but also on the choices of other players. The combination of all the players' choices is called a strategy profile. From an epistemic point of view, the strategy profile should include the beliefs players have about each other's possible actions (and beliefs). This model of interdependent decision making can essentially represent a wide array of social interactions. A metamodel for modelling epistemic aspects of a system as presented in (Rabbi et al., 2016a) is shown in Figure 5. Players in an epis-

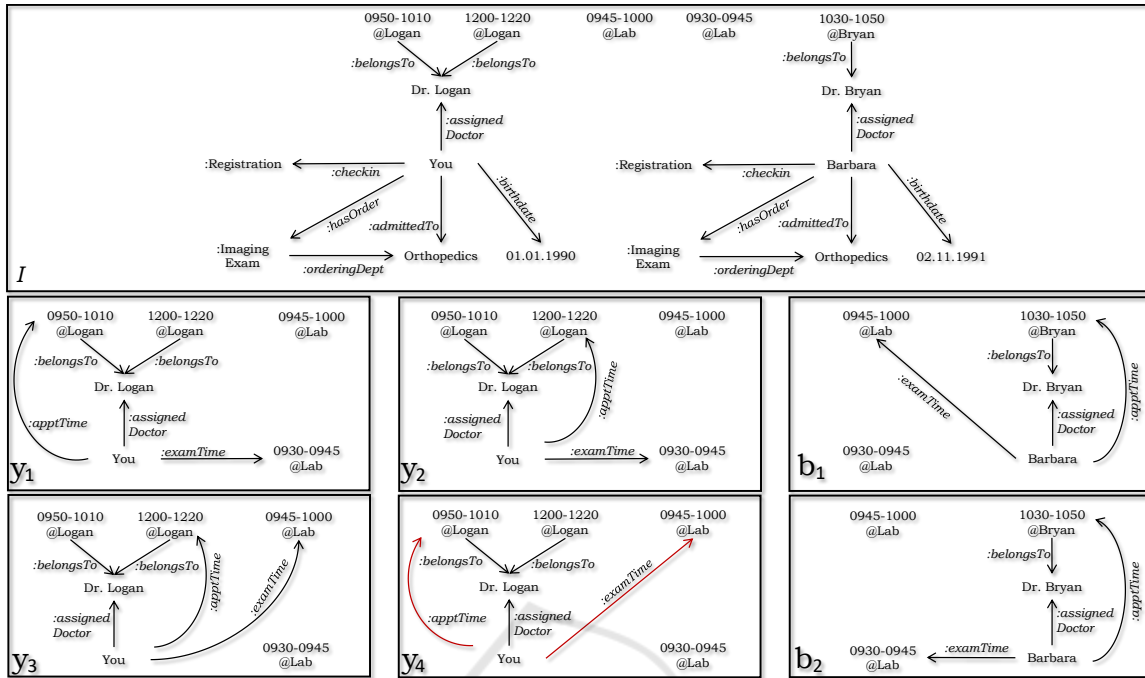


Figure 3: A model (I, I_1) of \mathcal{E}^+ (top) and individual resource allocation.

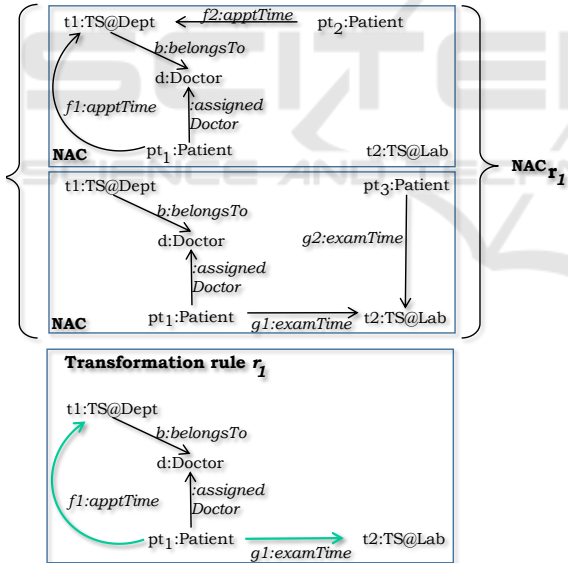


Figure 4: Transformation rule r_1 for individual resource allocation of patients.

temic game may have choices. The surjective constraint imposed on the *hasChoice* relation ensures that instances of *Choice* must be associated with players. An instance of *Belief* connects the choices of a player with other players' choices denoting the choice combination of players. In game theory, *utility* represents the motivation or satisfaction of players. The higher the utility the more the outcome is preferred. We use

the $\langle utility(n) \rangle$ predicate to assign utility to an instance of a belief. A utility assigned to an instance of a belief denotes the utility that a player derives from the outcome induced by the choice combination. We use the $\langle prob(r) \rangle$ predicate to annotate an instance of a belief with probability. Of course in a healthcare setting, patients do not have to know about each others choices; they are not expected to make decisions for mutual benefits. We propose to construct a strategy profile for them indirectly with an aim to optimize resource allocation respecting patients' preferences.

Given an initial epistemic model ξ of the meta-model in Figure 5 with two instances of *Player*: *You* and *Barbara*, we use coupled model transformation rules to automatically produce choices for you and Barbara. For each valid choice of resource allocation of individual patients from Figure 3, we create choices in ξ . The special kind of coupled transformation rule with coordinating edges creates choices ξ and establishes links from choices to instances of the merged metamodel \mathcal{E}^+ . Coordinating edges are used to co-ordinate between heterogeneous models and or modelling elements (Rabbi et al., 2014) (Rabbi et al., 2015). Figure 6 shows a transformation rule that identifies a player and produces choices for the player. The rule specifies that if we have an instance x of type *Player* in the epistemic model and x is an instance of *Patient* in a model (I, I_1) of \mathcal{E}^+ , such that x is allocated with an appointment time-slot z and an exam time-slot y , then we create an epistemic choice ω for player x

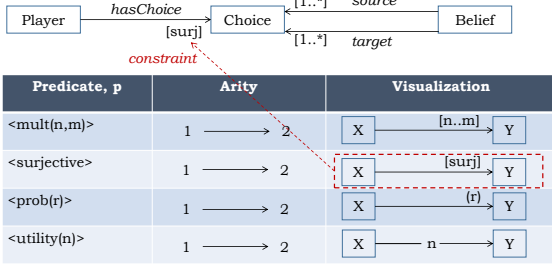
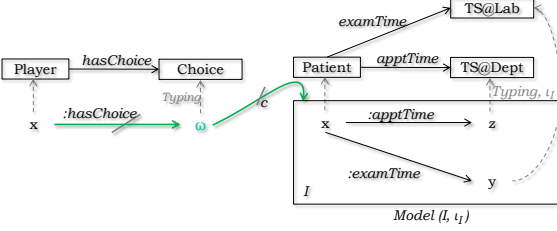


Figure 5: Metamodel for modelling epistemic aspects.


 Figure 6: Transformation rule r_2 for producing epistemic choices.

in the epistemic model and create a coordinating edge c linking the choice ω to the model (I, τ) .

Figure 7 illustrates the result of the application of this transformation rule over the possible choices of resource allocation from Figure 3. Figure 7(left) shows an epistemic model ξ where *You* and *Barbara* are instances of *Player*. Choices for you and Barbara are produced in ξ where the choices are linked to the models of the merged metamodel \mathfrak{S}^+ in Figure 7(right) by coordinating edges shown in blue. Note that we only produce epistemic choices for valid models of the entity metamodel. Since y_4 is not a valid model, we did not produce any epistemic choice for y_4 .

Figure 8 briefly presents the coordinating metamodel used for coordinating heterogeneous metamodelling hierarchies. The coordinating metamodel specifies that we may have *coordinating-edges* between nodes. The coordinating-edge may be instantiated in an arbitrary level of a metamodelling hierarchy. Rossini et al. presented a formal way of specifying this requirement in DPF using deep metamodelling (Rossini et al., 2012). A reference with multipotency at a metalevel i can be instantiated in all metalevels below i . Since the coordinating-edge has multi-potency, it can be instantiated in an arbitrary metalevel. In Figure 8, \mathfrak{S}^+ , y_1 , y_2 are placeholder nodes representing the identity of the (meta)models. The instance of the coordinating-edge c_1 establishes a link between an epistemic choice instance to the instance y_1 by referring to the node y_1 .

4 OPTIMIZATION OF RESOURCE ALLOCATION

In this section we discuss how resource allocation of distributed systems may be optimized using epistemic game theory. In our example, the patients were asked to come to the hospital at 9am and the clinic at the orthopedic department remains open until 5pm. Therefore, the maximum time spent by the patients to get healthcare services could be 8 hours (480 minutes). We calculate the utility based on the time spent at the hospital by the patients to get healthcare services. If a patient spends m minutes in total to get services, then her utility is $(480 - m)$. Therefore, the longer the patients are waiting, the lower utility they get. Our target is to optimize the resource allocation for the patients so that their wait time will be reduced. Consider the choice y_1 for you, which represents the resource allocation of 0930 – 0945@*Lab* and 0950 – 1010@*Logan*. This gives a utility of $480 - 70 = 410$ as you are spending only 70 minutes at the hospital to get your services. Similarly, we can calculate Barbara and your utilities and complete the epistemic model ξ with belief instances by performing a Cartesian product of Barbara and your choices as shown in Figure 9. Curved arrows in Figure 9 are representing belief instances which will be referred to as belief arrows. The belief arrow from y_1 to b_1 with an attribute 410 is representing an instance of *Belief* with utility 410.

An essential element of epistemic game theory analysis is the notion of belief hierarchies which are used to characterize solution concepts. We obtain belief hierarchies from an epistemic model by following belief arrows starting from any choice of a player. In our epistemic model we are only considering simple belief hierarchies. The idea of a simple belief hierarchy states that the whole belief hierarchy may be summarized by a combination of beliefs about players' choices. A belief hierarchy is simple if it contains at most one belief of each player. The epistemic model in Figure 9 represents six simple belief hierarchies of which three are: (i) $y_1 \xrightarrow{410} b_1 \xrightarrow{370} y_1$; (ii) $y_1 \xrightarrow{410} b_2 \xrightarrow{370} y_1$; (iii) $y_2 \xrightarrow{280} b_1 \xrightarrow{370} y_2$. Typically a belief hierarchy specifies a player's belief over other players' beliefs but in the healthcare context we use a belief hierarchy to specify a systems belief about patient's choices. The belief hierarchy $y_1 \xrightarrow{410} b_1 \xrightarrow{370} y_1$ can be read as: the system believes that you will consider making appointments represented in y_1 if Barbara considers making appointments represented in b_1 ; and Barbara considers making appointments represented in b_1 if you consider making appointments

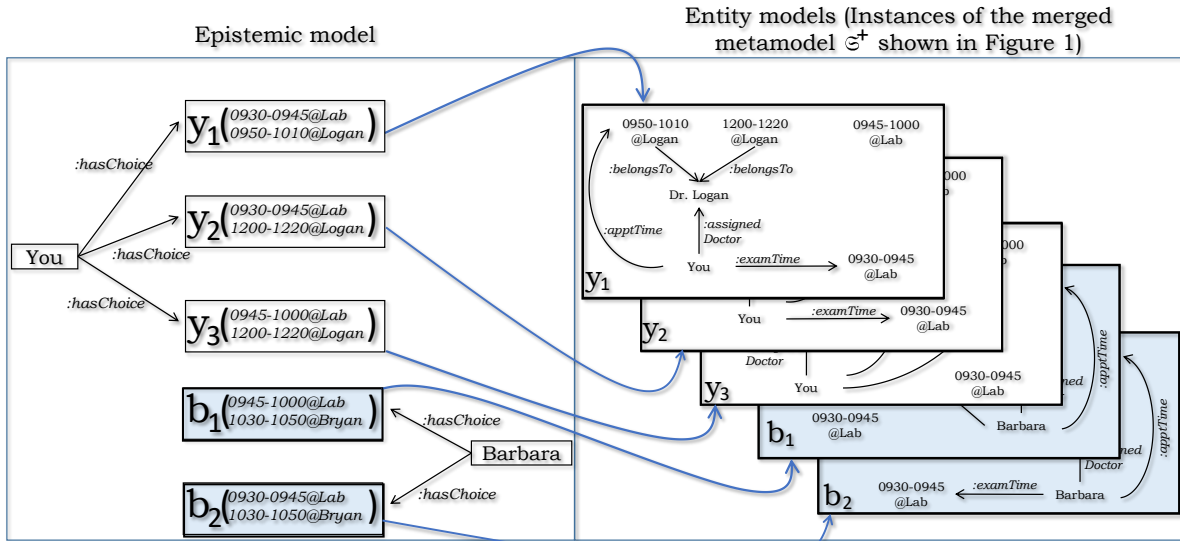


Figure 7: Epistemic choices produced by the application of model transformation rules.

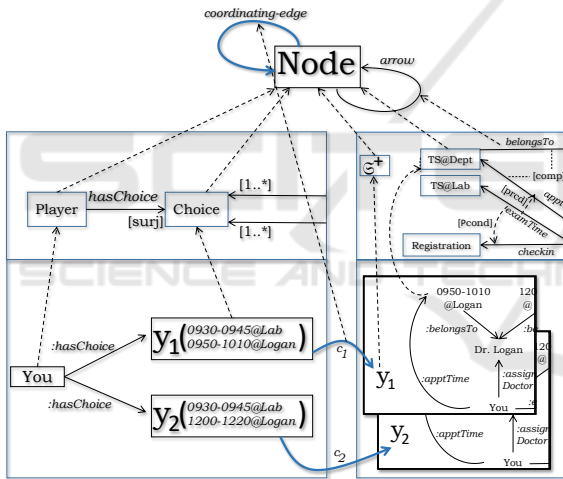


Figure 8: Heterogeneous metamodeling with Coordinating metamodel.

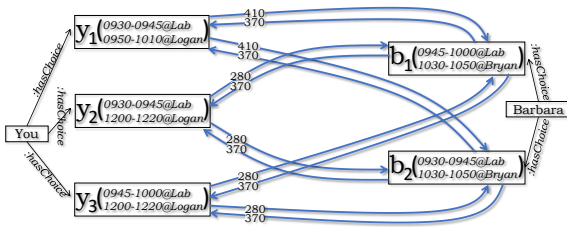


Figure 9: Epistemic model with combination of beliefs.

represented in y_1 . In our approach, uncertainty can be modelled by assigning probability to the belief arrows.

An important concern regarding the belief hier-

archies is their consistency. A consistent belief hierarchy represents a solution concept that satisfies all the constraints specified in the metamodel specifications. Following (Rabbi et al., 2016a) we determine how inconsistent combination of beliefs can be automatically removed from an epistemic model. If a combination of beliefs includes choices from a homogeneous metamodel, then the models are merged into a single model. For instance if we wish to check if the combination of beliefs represented by the belief hierarchy $y_1 \xrightarrow{410} b_1 \xrightarrow{370} y_1$ is consistent, we need to merge the models linked to y_1 and b_1 since they are typed by the merged metamodel \mathfrak{E}^+ . The merged model is then checked for consistency and if it violates any constraint specified in its metamodel, we conclude that the combination of beliefs is inconsistent and should be discarded from the epistemic model.

The combination of beliefs represented by the belief hierarchy $y_1 \xrightarrow{410} b_2 \xrightarrow{370} y_1$ is inconsistent as the merged instance is violating constraint **C7** (see section 2). The reason is that according to the injective constraint imposed on the reference *examTime*, patients cannot have the same time-slot. Therefore, assigning time-slot 0930 – 0945@Lab to both of you and Barbara is making the merged instance inconsistent. To check that a constraint is satisfied in a given model of a metamodel, we inspect the part of a model which is affected by the constraint. This is checked by projecting out the part of the model which is affected by the constraint. This is formally defined as a pullback operation in category theory.

Similarly we can show that the combination of beliefs represented by the belief hierarchies $y_2 \xrightarrow{280} b_2 \xrightarrow{370} y_2$, $y_3 \xrightarrow{280} b_1 \xrightarrow{370} y_3$ are inconsistent. Figure 10 shows an epistemic model with only consistent combination of beliefs.

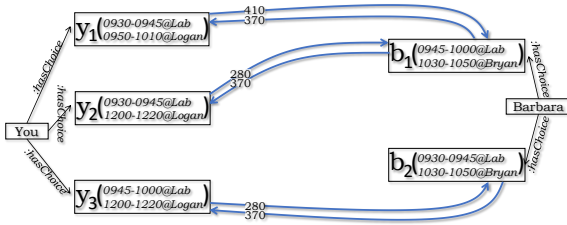


Figure 10: Epistemic model with consistent combination of beliefs.

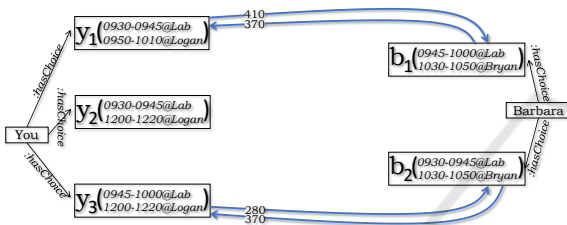


Figure 11: Epistemic model with rational combination of beliefs.

The epistemic choice y_2 which represents the resource allocation of 0930 – 0945@Lab and 1200 – 1220@Logan is not a rational choice since this is not optimal for any of the epistemic choices for Barbara. If you are allocated with time-slot 0930 – 0945@Lab at the radiology department, it will be optimal for you to see the doctor in appointment time-slot 0950 – 1010@Logan. Therefore, the epistemic choice y_2 is dominated by y_1 . The epistemic choice y_3 is a rational choice since this is optimal if Barbara is allocated with time-slot 0930 – 0945@Lab at the radiology department.

Figure 11 shows belief combinations with rational choices. The combination of beliefs is rational as the choices consist of the belief combinations are all rational choices with respect to the choice combination represented by the combination of beliefs.

Following (Perea, 2012) (Rabbi et al., 2016a), a simple belief hierarchy generated by a combination of beliefs leads to a *Nash equilibrium* if the combination of beliefs is rational. Therefore, we obtain two *Nash equilibria* from the epistemic model represented in Figure 11:

- Allocating time-slot 0930 – 0945@Lab and 0950 – 1010@Logan for you is rational if time-slot 0945 – 1000@Lab and 1030 – 1050@Bryan

are allocated to Barbara gives a total utility of $(410 + 370) = 780$;

- Allocating time-slot 0945 – 1000@Lab and 1200 – 1220@Logan for you is rational if time-slot 0930 – 0945@Lab and 1030 – 1050@Bryan are allocated to Barbara gives a total utility of $(280 + 370) = 650$.

However, the first equilibrium has a higher total utility for the system.

5 COORDINATION OF TRANSFORMATION RULES

Scalability is an important issue regarding the application of optimization techniques in real-life scenario. In our approach, we proposed to use model transformation rules for allocating resources. In general, the result of the application of a model transformation rule may become inconsistent i.e., may violate the constraints specified in a metamodel. Therefore, the application of a model transformation rule requires a consistency checking which is a time consuming operation. To tackle this problem, we need techniques to reduce the complexity of consistency checking. In this section, we introduce the idea of a compliant rule that preserves the validity of a model.

The transformation rule r_1 presented in Figure 4 is not compliant with the set of atomic constraints specified in the merged metamodel specification \mathfrak{S}^+ of Figure 1. If the rule is applied over a valid model, it does not guarantee that the result will be a valid model conforming to \mathfrak{S}^+ . The rule can be enhanced so that while matching with a model it makes sure that the result will be a valid model.

Definition 4 (Compliant Rule). *Given a metamodel specification $\mathfrak{S} = (S, C^{\mathfrak{S}} : \Sigma)$. A transformation rule r is compliant with a set of atomic constraints from $C^{\mathfrak{S}}$ if the application of r over any valid model of \mathfrak{S} results in a valid model of \mathfrak{S} .*

Figure 12 shows a transformation rule r_3 which is compliant with the set of atomic constraints $C^{\mathfrak{S}^+}$ specified in \mathfrak{S}^+ (Figure 1). The addition of new instances of *examTime* and *apptTime* in a valid model of \mathfrak{S}^+ can possibly violate atomic constraints **C3**, **C5**, **C6**, **C7** and **C8**. The transformation rule r_3 makes sure that when applied on a valid model of \mathfrak{S}^+ , the addition of new elements does not violate any of the above mentioned constraints. The additional matching condition $endTime(t2) < endTime(t1)$ specified in r_3 makes sure that **C8** is not violated. Therefore the application of r_3 will not require any further consistency checking.

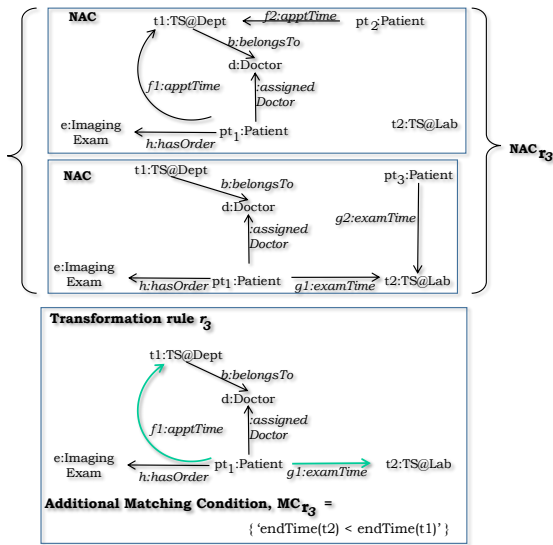


Figure 12: Transformation rule r_3 for individual resource allocation of patients.

Further enhancement of the rule is performed by amalgamation of rule r_3 (Figure 12) and r_2 (Figure 6). Rule amalgamation is a model transformation technique commonly used for coordinating rules with overlapping parts (Lamo et al., 2013). To apply a rule over a source model requires to perform an injective matching. In order to reduce the number of matching we coordinate rule r_3 and r_2 . Figure 13 shows a transformation rule r_4 that we formulate by combining r_3 and r_2 . While allocating resources for individual patients the rule r_4 produces epistemic choices at the same time. The rule is also compliant with the set of constraints specified in the merged metamodel specification and therefore if the rule is applied over a valid model, it guarantees consistency.

6 RELATED WORK

The necessity of merging different formalisms of software specifications is intrinsic to the discipline of software engineering. Category theory has been considered to be a viable solution to accommodate the diversity of formalisms (Fiadeiro and Maibaum, 1995; Diskin et al., 2010; Rutle, 2010). Diskin et al., (Diskin et al., 2010) introduced the concept of inter-metamodel constraints to specify constraints over interdependent models and later on König et al., (König and Diskin, 2016) proposed an advanced consistency checking algorithm for inter-metamodel constraints. A slight different but related idea was presented by Rabbi et al., in (Rabbi et al., 2014) which formalizes the co-ordination among metamodels, using a lingu-

istic extension of the metamodelling hierarchy. The proposed linguistic extension is based on an added metalevel which models the integration of two or more different aspects of a system.

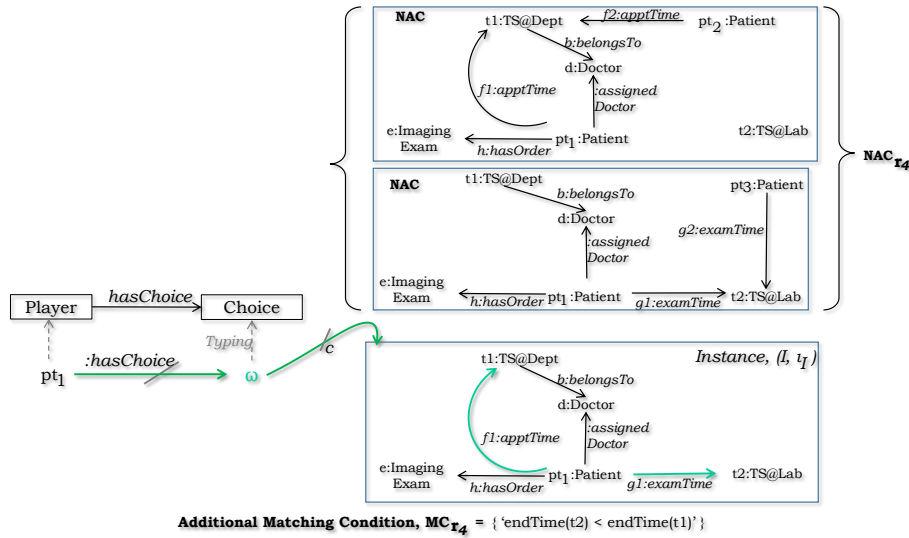
Bak et al. presented a language called Clafer that unifies features and class modelling (Bak et al., 2016). The language is supported by a tool that supports model based analyses such as consistency checking, instance completion, single and multi-objective optimization. The tool utilizes existing model based analyzers such as Alloy (relational logic solver) (Jackson, 2002), Z3 (SMT solver) (De Moura and Bjørner, 2008), Choco 3 (constraint satisfaction problems and constraint programming solver) (Jussien et al., 2008) by translating Clafer models into the backend solvers. While Clafer provides a concise notation for feature modeling it does not support modelling distributed systems with inter-metamodel constraints. Also the optimization technique is different in contrast to the optimization technique of epistemic game theory.

Alanen and Porres investigated an approach for automatic discovery of the difference and union of models in the context of a version control system (Alanen and Porres, 2003). They presented algorithms to calculate the difference between models for MOF-based models. This work can be adapted in order to identify overlaps between component models for merging DPF (meta)model specifications.

7 CONCLUSION

This paper is based on the metamodelling epistemic game theory approach introduced in (Rabbi et al., 2016a). Epistemic games are relevant for modeling distributed systems in order to optimize the use of shared resources. We proposed several enhancement in this paper by incorporating model merge with inter-metamodel constraints and compliant rules.

In this paper, we adopted merged metamodel specifications and specified distributed resource constraints via inter-metamodel constraints. We applied model transformations to produce epistemic game theory models for the purpose of optimizing the use of distributed resources. Currently we are investigating the Clafer (Bak et al., 2016) and WebDPF tool (Rabbi et al., 2016b) for reasoning over DPF metamodels. In order to apply the proposed method efficiently, we presented the concept of compliant rules. An automatic resolution of compliant rules is out of scope of this paper and it is currently under investigation.


 Figure 13: Transformation rule r_4 for resource allocation and producing epistemic choices.

REFERENCES

- Alanen, M. and Porres, I. (2003). *Difference and Union of Models*, pages 2–17. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Bak, K., Diskin, Z., Antkiewicz, M., Czarnecki, K., and Wasowski, A. (2016). Clafer: unifying class and feature modeling. *Software and System Modeling*, 15(3):811–845.
- Barr, M. and Wells, C., editors (1995). *Category Theory for Computing Science, 2nd Ed.* Prentice Hall International (UK) Ltd., Hertfordshire, UK.
- De Moura, L. and Bjørner, N. (2008). Z3: An efficient smt solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'08/ETAPS'08*, pages 337–340, Berlin, Heidelberg. Springer-Verlag.
- Diskin, Z. (2011). *Model Synchronization: Mappings, Tiles, and Categories*, pages 92–165. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Diskin, Z. and Wolter, U. (2008). A diagrammatic logic for object-oriented visual modeling. *Electronic Notes in Theoretical Computer Science*, 203(6):19 – 41. Proceedings of the Second Workshop on ACCAT, 2007.
- Diskin, Z., Xiong, Y., and Czarnecki, K. (2010). Specifying overlaps of heterogeneous models for global consistency checking. In *Proceedings of the First International Workshop on Model-Driven Interoperability, MDI '10*, pages 42–51, NY, USA. ACM.
- Ehrig, H., Ehrig, K., Prange, U., and Taentzer, G. (2006). *Fundamentals of Algebraic Graph Transformation*. Monographs in Theoretical Computer Science. Springer.
- Fiadeiro, J. L. and Maibaum, T. (1995). Interconnecting formalisms: Supporting modularity, reuse and incrementality. *SIGSOFT Softw. Eng. Notes*, 20(4):72–80.
- Han, P. K., Klein, W. M., and Arora, N. K. (2011). Varieties of uncertainty in health care: A conceptual taxonomy. *Medical decision making: an international journal of the Society for Medical Decision Making*, 31(6):828–838.
- Jackson, D. (2002). Alloy: A lightweight object modelling notation. *ACM Trans. Softw. Eng. Methodol.*, 11(2):256–290.
- Jussien, N., Rochart, G., and Lorca, X. (2008). Choco: an Open Source Java Constraint Programming Library. In *CPAIOR'08 Workshop on Open-Source Software for Integer and Constraint Programming (OSSICP'08)*, pages 1–10, Paris, France, France.
- König, H. and Diskin, Z. (2016). Advanced local checking of global consistency in heterogeneous multimodeling. In Wasowski, A. and Lönn, H., editors, *Modelling Foundations and Applications - 12th European Conference, ECMFA'16/STAF'16*, volume 9764 of LNCS, pages 19–35. Springer.
- Lambers, L., Ehrig, H., Prange, U., and Orejas, F. (2008). *Embedding and Confluence of Graph Transformations with Negative Application Conditions*, pages 162–177. Springer Berlin Heidelberg.
- Lamo, Y., Mantz, F., Rutle, A., and de Lara, J. (2013). A declarative and bidirectional model transformation approach based on graph co-spans. In *Proceedings of the 15th Symposium on Principles and Practice of Declarative Programming, PDPD '13*, pages 1–12, NY, USA. ACM.
- Mantz, F. (2014). *Coupled Transformations of Graph Structures applied to Model Migration*. PhD thesis, Department of Mathematics and Informatics, Philipps-Universität Marburg, Germany.
- Perea, A. (2012). *Epistemic Game Theory: Reasoning and Choice*. Cambridge University Press, 1 edition.

- Perea, A. (2014). From classical to epistemic game theory. *International Game Theory Review*, 16(01):1440001.
- Pinelle, D. and Gutwin, C. (2006). Loose coupling and healthcare organizations: Deployment strategies for groupware. *Comput. Supported Coop. Work*, 15(5-6):537–572.
- Rabbi, F., Lamo, Y., and MacCaull, W. (2014). Co-ordination of multiple metamodels, with application to healthcare systems. In *1st International Workshop on (Meta)modelling for Healthcare Systems (MMHS), 2014*, volume 37 of *Procedia Computer Science*, pages 473–480. Elsevier.
- Rabbi, F., Lamo, Y., and Yu, I. C. (2016a). Towards a categorical approach for meta-modelling epistemic game theory. In *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*, pages 57–64. ACM.
- Rabbi, F., Lamo, Y., Yu, I. C., and Kristensen, L. M. (2015). Towards a Multi Metamodelling Approach for Developing Distributed Healthcare Applications. . *NIK: Norsk Informatikkonferanse*, ISSN: 1892-0721.
- Rabbi, F., Lamo, Y., Yu, I. C., and Kristensen, L. M. (2016b). WebDPF: A web-based metamodelling and model transformation environment. In *MODELSWARD 2016 - Proceedings of the 4rd International Conference on Model-Driven Engineering and Software Development, Rome, Italy, 19-21 February, 2016.*, pages 87–98. SciTePress.
- Rossini, A., de Lara, J., Guerra, E., Rutle, A., and Lamo, Y. (2012). A graph transformation-based semantics for deep metamodelling. In *Proceedings of the 4th International Conference on Applications of Graph Transformations with Industrial Relevance*, AGTIVE'11, pages 19–34. Springer-Verlag.
- Rutle, A. (2010). *Diagram Predicate Framework: A Formal Approach to MDE*. PhD thesis, Department of Informatics, University of Bergen, Norway.
- Wang, X., Rutle, A., and Lamo, Y. (2015). Towards user-friendly and efficient analysis with alloy. In *Proceedings of the 12th Workshop on Model-Driven Engineering, Verification and Validation, MoDeVva@MoDELS 2015*, volume 1514 of *CEUR Workshop Proceedings*, pages 28–37. CEUR-WS.org.