

Hybrid Class Diagrams

Mira Balaban, Azzam Maraee, Gera Weiss

Computer Science Department
Ben-Gurion University of the Negev, Beer-Sheva 84105, ISRAEL
`mira,mari,geraw@cs.bgu.ac.il`

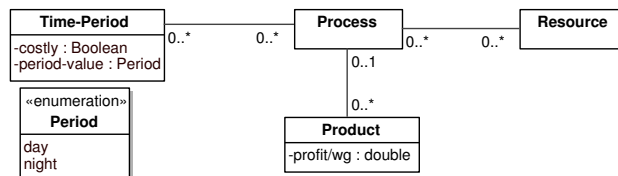
Abstract. A formal model of *hybrid class diagram* is suggested that allows for models with discrete and continuous entities. Mathematical measure theory is used for generalizing discrete multiplicity constraints. We show that typical inter-relationships between multiplicity constraints and linear inequalities are preserved. The semantics of the hybrid model are defined as a direct generalization of the semantics of the traditional discrete class diagram model, where general measures are used to generalize counting.

1 Introduction

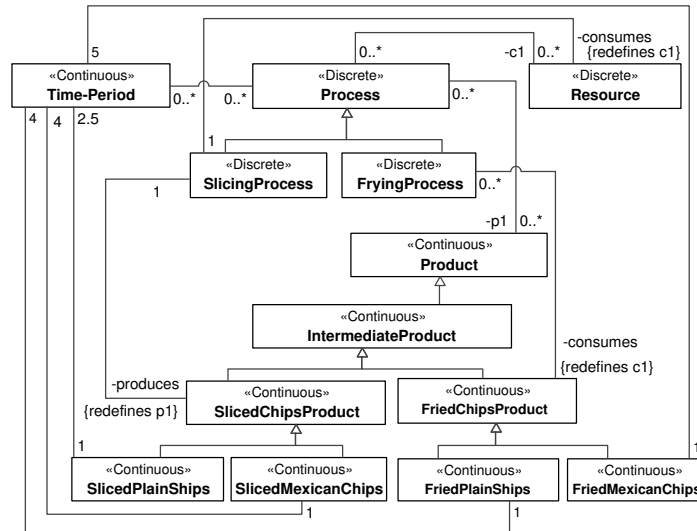
Modern information systems cope with standard problems of software engineering, such as ever growing complexity, technology evolution, constant requirement changes, dynamic software architecture, and more. The design of such systems requires multiple levels of abstraction, that can be obtained by using a *Model Driven Engineering (MDE)* approach, in which software is developed by repeated transformations of models.

In many application areas, these techniques need to be applied to hybrid domains where both continuous and discrete quantities are to be considered. Similar to hybrid automata, where discrete and continuous dynamics may interact in nontrivial ways, data models may also contain nontrivial connections between continuous and discrete quantities. As the language of *class diagrams* forms the backbone of modeling relations between entities, we propose hybrid class diagrams for modeling systems that contain both discrete and continuous sets. The need for hybrid class diagrams is best explained with an example:

Example 1 Consider a company for producing chips. The production includes several processes, each with additional specializations, based on the kind of chips that is produced. Process activation is costly, and is affected by the activation period during day or night. A reasonable initial model (following the visitor design pattern) is:



Later on, the model is further specialized, to account for concrete production details. In order to determine an optimal production plan, the company needs to consider multiplicity constraints between time periods and products, e.g., “slicing a kilogram of potatoes for plain chips takes two and a half hours, while slicing for Mexican chips takes four hours”, “frying for plain chips takes four hours, while frying for Mexican chips takes five hours”, and profit constraints such as “The profit from one kilogram of plain chips is \$1.5”. A somewhat detailed refinement of the class diagram that includes these constraints can be:



Such problems can be solved by standard mathematical programming methods, that operate over continuous domains, producing solutions that require, say, 8.75 night hours for producing 1050.4 kilograms of Mexican chips. Using these methods implies that some classes, e.g., the *TimePeriod*, *SlicedPlainChips* and *SlicedMexicanChips*, denote continuous sets! In the figure they are marked by a stereotype **Continuous**, while the rest are stereotyped as **Discrete**.

Using continuous methods for determining class proper instantiation affects model design and instantiation. For example, if the mathematical methods find that there is no solution to the given constraints, then there might be a need for model revision; if a solution is found then its realization requires creation of an instance, or checking the integrity of created instances. Therefore, the meaning of a *hybrid class diagram* that might include both continuous and discrete classes must be formally defined.

In this paper we suggest a *hybrid class diagram* model, that bridges between discrete and continuous classes. We formulate the semantics of continuous classes and of cardinality constraints between continuous classes, or between a discrete and a continuous class. Furthermore, we show that a solution to inequalities that optimize some parameters in the problem domain, corresponds to a legal instance of the hybrid class diagram. The novelty of our approach is that it naturally

generalizes the traditional discrete class diagram model, so that discrete class diagrams and their constraints become a special case of the hybrid class diagram model.

Section 2 introduces the hybrid class diagram model. Section 3 shows that typical inter-relationships between multiplicity constraints and linear inequalities are preserved. Section 4 summarizes related work, and Section 5 describes our plans for future research.

2 Hybrid Class Diagrams

In this section we present the syntax and the semantics of *hybrid class diagram* using a generalization of the way UML (discrete) class diagrams are usually defined.

2.1 Syntax of hybrid class diagrams

For clarity of presentation, we focus only on binary associations, leaving more involved constructs such as n -ary associations and hierarchies for the full version of the paper. This choice is made because the unique characteristics of hybrid class diagrams, as compared to standard class diagrams, are visible already in this basic setting and because the other constructs do not interact with the hybridness of the class diagram.

Definition 1 A hybrid class diagram is a tuple $\langle \mathcal{C}, type, \mathcal{A}, \mathcal{R}, \rho, \gamma, Constraint \rangle$ where

- \mathcal{C} is a set of class symbols.
- $type: \mathcal{C} \rightarrow \{\text{discrete}, \text{continuous}\}$ is a mapping that assigns types to classes.
- \mathcal{A} is a set of association symbols.
- \mathcal{R} is a set of role symbols.
- $\rho: \mathcal{A} \rightarrow \mathcal{R} \times \mathcal{R}$ is a one-to-one assignment of (unique) roles to associations.
- $\gamma: \mathcal{R} \rightarrow \mathcal{C} \times (\mathbb{R} \cup \{*\}) \times (\mathbb{R} \cup \{*\})$ is an assignment of classes and multiplicity constraints to roles.

2.2 Semantics of hybrid class diagram

The direct semantics of hybrid class diagrams generalizes the direct semantics of UML class diagrams in defining issues of measuring the size of sets. While in discrete class diagrams, size is interpreted as cardinality, in hybrid class diagrams, a finer notion of size, based on *measure theory*, is employed.

An *instance* \mathcal{I} of a hybrid class diagram $\langle \mathcal{C}, type, \mathcal{A}, \mathcal{R}, \rho, \gamma, Constraint \rangle$ consists of a *domain* \mathcal{D} and an extension function I that assigns denotations (extensions) to symbols. The domain \mathcal{D} is a measure space $\langle X, \Sigma, \mu \rangle$ (see e.g. [1] for background on measure theory). For a class symbol C , C^I (a shorthand for $I(C)$) is a subspace of \mathcal{D} , and for an association symbol a , a^I is a subspace of the product measure space of \mathcal{D} . A *legal instance* is an instance that satisfies the class typing and constraints:

1. For every class symbol $C \in \mathcal{C}$:
 - (a) If $type(C) = \mathbf{discrete}$, C^I is isomorphic to a subset of \mathbb{N} with the *counting* measure.
 - (b) If $type(C) = \mathbf{continuous}$, C^I is isomorphic to a subset of \mathbb{R} with the Borel measure.
2. For every association symbol $a \in \mathcal{A}$ with $\rho(a) = \langle r_1, r_2 \rangle$, $\gamma(r_1) = \langle C_1, min_1, max_1 \rangle$, and $\gamma(r_2) = \langle C_2, min_2, max_2 \rangle$: For $i \neq j \in \{1, 2\}$, for almost every entity $\mathbf{e} \in C_i^I$, $a_{r_j}^I(\mathbf{e})$ is measurable, and $\mu(a_{r_j}^I(\mathbf{e})) \in [min_j, max_j]$.

Some clarifications for this definition are in place:

- The term “almost every” in part 2 means “all but a set of measure zero” as common in measure theoretic literature.
- The proposed semantics of multiplicity constraints is a direct generalization of standard semantics for (non hybrid) class diagrams. Specifically, when all classes are discrete, the definitions are equivalent.
- Relying on the fact that the classical semantics of multiplicity constraints dictate that, for any $\mathbf{E} \subseteq C_i^I$,

$$min_j \cdot |\mathbf{E}| \leq \sum_{\mathbf{e} \in \mathbf{E}} |a_{rn_j}(\mathbf{e})| \leq max_j \cdot |\mathbf{E}|,$$

restriction 2 in the above definition can be replaced with a requirement that

$$min_j \cdot \mu(\mathbf{E}) \leq \int_{\mathbf{e} \in \mathbf{E}} \mu(a_{rn_j}(\mathbf{e})) d\mu \leq max_j \cdot \mu(\mathbf{E})$$

for all $\mathbf{E} \subseteq C_i^I$. This would give an alternative definition that may appeal to some because it only refers to sets and not to individual elements (bypassing the need to clarify the meaning of an individual element of a hybrid set). Since it is an easy exercise in measure theory to prove that the two definitions are equivalent, choosing between them is a matter of taste.

- One can visualize a^I as a subspace of the product $C_1^I \times C_2^I$. Then, restriction 2 can also be stated as requiring that

$$min_j \cdot \mu(\mathbf{E}) \leq \mu^2(\{\langle \mathbf{e}, \mathbf{e}' \rangle : \langle \mathbf{e}, \mathbf{e}' \rangle \in a^I \cap E \times C_j^I\}) \leq max_j \cdot \mu(\mathbf{E})$$

for all $\mathbf{E} \subseteq C_i^I$ where μ^2 is the product measure specifying “area” in the product space.

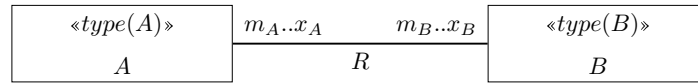
3 Instantiation of Hybrid Class Diagrams

In this section we solve the problem of instantiation of hybrid class diagrams. We focus on instantiating a single binary association because, once this is solved, a solution for a more involved class diagrams can be deduced similar to the way it is done with classical (non-hybrid) class diagrams (see e.g., [2, 3]).

Formally, for a binary association, we show how to translate a solution of the inequalities that are standardly associated with the multiplicity constraints to an instance that conforms with the solution. As the reverse direction is trivial, it shows that the inequalities capture the constraints imposed by the class diagram and that instances of the class diagram are in one-to-one correspondence with the solutions of the inequalities. One possible application of this fact can be a method for finding “best” instances of a class diagram using a linear programming solver that takes the inequality and some optimality measure and returns a solution that can be translated to an instance.

A formal problem of the mathematical problem that we solve in this section is given as follows:

Problem 1 (Instantiation of Hybrid Class Diagrams) *Given a hybrid class diagram*



(where $type(A), type(B) \in \{\mathbf{discrete}, \mathbf{continuous}\}$) and numbers r, a, b that solve the inequalities

$$m_B \cdot a \leq x_B \cdot a, \quad m_A \cdot b \leq x_A \cdot b, \quad r \leq a \cdot b \quad (*)$$

find a legal instance, I , for the class diagram such that

$$\mu(A^I) = a, \quad \mu(B^I) = b, \quad \mu^2(R^I) = r. \quad (**)$$

We are going to propose solutions to this problem in incremental steps. First, we will recall known solutions to the problem when $type(A) = type(B) = \mathbf{discrete}$. Later, we will generalize the solution to the case that both classes are continuous and, then, to the case where one of the classes is discrete and the other is continuous.

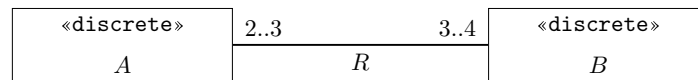
3.1 The Discrete Case: $type(A) = type(B) = \mathbf{discrete}$

Solutions to this case have been proposed in [2, 4]. These solutions can be viewed as going cyclically over the set $A^I \times B^I$ in a diagonal direction. Specifically, if we index the elements of A^I as a_1, a_2, \dots, a_n and the elements of B^I as b_1, b_2, \dots, b_m , the pairing is given by

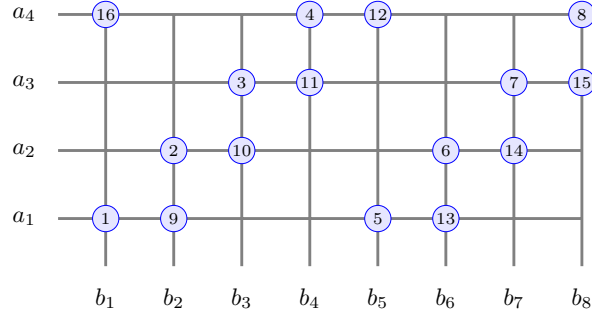
$$R^I = \{\langle a_1, b_1 \rangle, \langle a_2, b_2 \rangle, \dots, \langle a_{i \bmod n}, b_{i \bmod m} \rangle, \dots, \langle a_{r \bmod n}, b_{r \bmod m} \rangle\}.$$

This is best illustrated by an example, as follows.

Given an association



and a solution $r = 16, a = 4, b = 8$ for the inequalities (*), a legal instance I that satisfies (**) is illustrated in the following diagram:



In this instance, $A^I = \{a_1, \dots, a_4\}$, $B^I = \{b_1, \dots, b_8\}$ and R^I is represented by the circled coordinates (the numbers depict the order at which the elements of R^I are constructed).

This graphical depiction hints to how we are going to generalize the construction to continuous classes. Visually, instances consist of diagonal “stripes” that flow upwards in 45° and wrap around when they hit the boundary, similar to the way, in the well known Snake video game, the head of the snake continues from opposite side when it exists the screen (see e.g. [http://en.wikipedia.org/wiki/Snake_\(video_game\)](http://en.wikipedia.org/wiki/Snake_(video_game))). We will apply this intuition also to continuous sets, below.

3.2 The Continuous Case: $\text{type}(A) = \text{type}(B) = \text{continuous}$

We now turn to study the case where both classes are continuous. Before presenting two different constructions and proving their correctness, we note a lemma that simplifies the exposition.

Looking at the solution to the discrete case, presented in the preceding section, one may note that the instance is chosen such that the pairing of elements of A to elements of B is almost uniform, i.e., the number of B s paired to every member of A , and vice versa, varies by at most one.

Because r/a must fall in the interval $[m_B, x_B]$ and r/b must fall in the interval $[m_A, x_A]$, it turns out that, in the continuous case, that we can limit the discussion to finding instances that assign sets of exactly the same size (no variation), as stated by the following lemma:

Lemma 31 *For any a, b, r that satisfy (*), there is a legal instance I that satisfies (**) if and only if there is an instance I that satisfies (**) such that $\mu(R(\alpha)) = r/a$ for all $\alpha \in A^I$ and $\mu(R(\beta)) = r/b$ for all $\beta \in B^I$.*

Here (and from now on), $R(\alpha)$ denotes the set of B elements associated with α and $R(\beta)$ denotes the set of A elements associated with β .

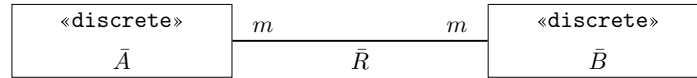
The lemma allows us to ignore the cardinality constraints of the class diagram and focus only on the numbers a, b and r . Specifically, instead of requiring that

the measure of the set of B 's associated with each member of A be between m_B and x_B , we, from now on, without loss of generality, require that it is exactly r/a . Similarly, we will require that the measure of the set of A 's associated with every member of B be r/b .

Based on this lemma, we now propose two constructions that solve the problem of instantiation of class diagrams in the case where both classes are continuous. We propose two constructions: the first goes by a reduction to the discrete case and works only when r/ab is rational, and the second is a direct construction that works even if r/ab is not rational. Note that the requirement that r/ab be rational can be circumvented by approximations, i.e., for practical purposes, we can always use a rational number that is close enough to r/b .

Construction by reduction to the discrete case: The first construction works when r/ab is a rational and goes by reduction to the discrete-case, in three steps, as follows:

1. Let r, a, b be a solution of (*) and let $m, n \in \mathbb{N}$, such that $r/ab = m/n$ (this is where we use the assumption that this term is rational).
2. Create a legal instance \bar{I} for the class diagram



with the solutions $\bar{r} = mn, \bar{a} = n, \bar{b} = n$ using, for example, the construction presented in Section 3.1.

3. Based on \bar{I} , construct an instance I for the continuous problem by

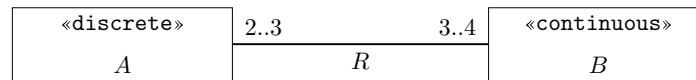
$$A^I = (0, a], \quad B^I = (0, b],$$

and

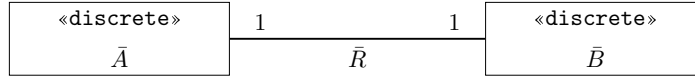
$$R^I = \bigcup_{\langle i, j \rangle \in \bar{R}^I} \left(\frac{ia}{n}, \frac{(i+1)a}{n} \right] \times \left(\frac{jb}{n}, \frac{(j+1)b}{n} \right].$$

The idea of this construction is to divide the box $A^I \times B^I$ to n^2 identical boxes each of which is similar to the whole box (same height/width ratio). Then choose R^I to be a union of some of the boxes. We call this a reduction because the selection of boxes can be done using the construction presented in Section 3.1 above.

We demonstrate how the construction works by an example: consider a class diagram



with two continuous classes and the solution $a = 4, b = 8, r = 16$ of (*). Since $r/ab = 1/2$, we have $m = 1$ and $n = 2$. To find an instance for this class diagram, we first solve for the class diagram



(where both classes are discrete) with $\bar{a} = \bar{b} = n = 2$ and $\bar{r} = mn = 2$. Applying the construction of Section 3.1 to this simple case gives $\bar{R}^I = \{\langle a_1, b_1 \rangle, \langle a_2, b_2 \rangle\}$. According to the third step above, we get $R^I = (0, 2] \times (0, 4] \cup (2, 4] \times (4, 5]$ which is a union of the two boxes on the positive diagonal.

Because \bar{R}^I associates the members of \bar{A} with the members \bar{B} uniformly (same number of associations to each member), and because pairs $\langle i, j \rangle \in \bar{R}^I$ are translated to identical non-overlapping boxes, it is easy to verify that R^I maps the elements of A and B uniformly. Therefore, by Lemma 31, we get:

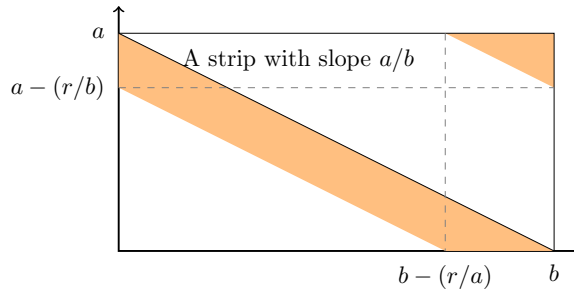
Proposition 32 *For a hybrid class diagram as in Problem 1, where $\text{type}(A) = \text{type}(B) = \text{continuous}$, and a solution r, a, b of (*) that satisfies $r/ab \in \mathbb{Q}$, the above construction gives a legal instance that satisfies (**).*

We continue with a presentation of an alternative, direct, construction that does not depend on r/ab being rational nor on a solution to the discrete case.

A direct construction for continuous-continuous relations: For a, b, r that solve (*), an instance I is given by $A^I = (0, a]$, $B^I = (0, b]$, and $R^I = R_1 \cup R_2 \cup R_3$ where

$$\begin{aligned}
 R_1 &= \left\{ x \in A^I \times (0, b - \frac{r}{a}] : x \in (a - \frac{a}{b}x_1 - \frac{r}{b}, a - \frac{a}{b}x_1] \right\}, \\
 R_2 &= \left\{ x \in A^I \times (b - \frac{r}{a}, b] : x \in (0, a - \frac{a}{b}x_1] \right\}, \\
 R_3 &= \left\{ x \in A^I \times (b - \frac{r}{a}, b] : x \in (2a - \frac{a}{b}x_1 - \frac{r}{b}, a] \right\}.
 \end{aligned}$$

The following diagram depicts a visual representation of this construction:



Note the similarity to the discrete case: here also we have a stripe that “wraps around” from the opposite side when hitting the boundaries of the box (like the Snake video game).

Based on Lemma 31, it is easy to verify that this is indeed the instance that we are looking for. All we need to verify is that the intersection of every

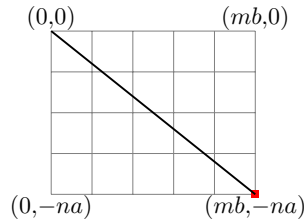
horizontal line with the shaded area is a collection of interval of total length r/a , and that the total length of the intersection of every vertical line with the shaded area is r/b . This proves the following proposition:

Proposition 33 *For a hybrid class diagram as in Problem 1, where $\text{type}(A) = \text{type}(B) = \text{continuous}$, and a solution r, a, b of $(*)$, the construction creates a legal instance that satisfies $(**)$.*

3.3 A generalization of the direct construction

In this section we generalize the direct construction presented in the previous section. The construction is also based on the definition of stripes in the box $(0, a] \times (0, b]$. We show that the instance presented before is a representative of a class of instances and present the formula for the other instances of this class. One reason to present all the instances is for future use in applications where further constraints are imposed. We, again, choose $A^I = (0, a]$, $B^I = (0, b]$ and construct R^I as follows:

1. Consider an arbitrary line l , which intersects the grid $a\mathbb{Z} \times b\mathbb{Z}$ at more than one point and which goes through the origin. Denote the slope of the line by s_A/s_B .
2. Let $(mb, -na)$, $n, m \in \mathbb{N}$ be the first intersection point of l with the grid $a\mathbb{Z} \times b\mathbb{Z}$ below the origin, as depicted in the diagram:

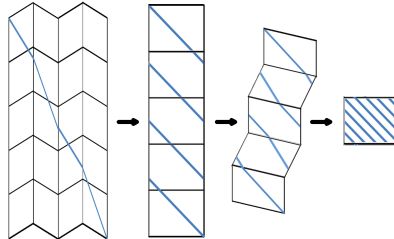


3. Define a stripe in \mathbb{R}^2 as follow:

$$\text{stripe} = \{(x_1, x_2) \in \mathbb{R}^2 : x_2 \in (a - \sigma x_1 - w, a - \sigma x_1]\}$$

where $\sigma = \frac{s_A}{s_B} = -\frac{na}{mb}$ is the slope of the line l and $w = \frac{r}{b} \cdot \frac{1}{m}$ is the vertical width of the stripe.

4. Let $\tilde{R} = \text{stripe}/\sim$, where \sim is the equivalence relation such that $(x_1, x_2) \sim (x_1 + ia, x_2 + jb)$ for all $i, j \in \mathbb{Z}$. The equivalence relation \sim can be visualized as folding along the grid lines and considering points that are at the same displacement from a grid point as equivalent, as depicted in the diagram:



The set \tilde{R} consists of the equivalence classes that intersect the stripe.

5. Choose $R^I = \{\tilde{r} \cap (A \times B) : \tilde{r} \in \tilde{R}\}$.

Given a line with slope s_A/s_B , the natural numbers m, n defined in the construction above can be computed by:

$$m = \frac{LCM(s_A \cdot b, s_B \cdot a)}{s_A \cdot b}, \quad n = \frac{LCM(s_A \cdot b, s_B \cdot a)}{s_B \cdot a}.$$

Proposition 34 *For a hybrid class diagram as in Problem 1 where $\text{type}(A) = \text{type}(B) = \text{continuous}$ and a solution r, a, b of (*), the construction gives a legal instance that satisfies (**) for any possible selection of the line l .*

Proof. By Lemma 31, we need to show that the construction creates an instance in which every vertical cut which goes through $(0, B]$ meets m stripes of width $r/b \cdot 1/m$ and every horizontal cut meets n stripes of height $r/b \cdot 1/n$. To show that it is sufficient to prove:

1. The line l intersects the vertical grid lines $\mathbb{R} \times a\mathbb{Z}$ in n non equivalent points under \sim and that the distance between two successive meeting points is equal. Similarly for the meeting with the horizontal grid lines.
2. The horizontal distance between every two lines is greater than r/an and the vertical distance is greater than r/bm . This means that we can expand the line into a stripe of width r/an and height r/bm .

The line l meets the vertical grid lines $\mathbb{R} \times a\mathbb{Z}$ in $\{(im/n)b, ia) : i = 1, \dots, n\}$. We are interested in the “modulo b ” part of the meetings:

$$\left\{ \left(\frac{imb}{n} \bmod b, ia \right) : i = 1, \dots, n \right\} = \left\{ \left(\frac{im \bmod n}{n} b, ia \right) : i = 1, \dots, n \right\}$$

Since m and n are relatively primes, from elementary modulo n algebra, we have that $\{m \bmod n, 2m \bmod n, \dots, nm \bmod n\} = \{0, 1, \dots, n-1\}$ and therefore

$$\left\{ \frac{im \bmod n}{n} \cdot b : i = 1, \dots, n \right\} = \left\{ 0, \frac{b}{n}, \frac{2b}{n}, \dots, \frac{(n-1)b}{n} \right\}.$$

Similarly, for the horizontal axis meetings. In total, the meeting points are:

$$\left\{ (0, 0), \left(\frac{b}{n}, 0 \right), \dots, \left(\frac{(n-1)b}{n}, 0 \right) \right\}, \left\{ (0, a), \left(\frac{b}{n}, a \right), \dots, \left(\frac{(n-1)b}{n}, a \right) \right\}$$

for the vertical lines, and

$$\left\{ (0, 0), \left(\frac{a}{m}, 0 \right), \dots, \left(\frac{(m-1)a}{m}, 0 \right) \right\}, \left\{ (0, b), \left(\frac{a}{m}, b \right), \dots, \left(\frac{(m-1)a}{m}, b \right) \right\}$$

for the horizontal lines.

The distance between each successive points in the vertical lines meetings and each successive points in the horizontal lines meetings are b/m and a/n ,

respectively. It remains to show that vertical distance between the vertical lines meetings is greater or equal to r/mb and is greater or equal to r/na for the horizontal intersections. We show for the vertical axis sides, the other direction is similar. The vertical distance between the meetings is b/m . The inequalities solution satisfies $r \leq ab$. Therefore, $r/b \cdot 1/m \leq a/m$. Note the special case where $r = ab$ (equality), in which case the whole box is filled, i.e., $R^I = (0, a] \times (0, b]$. \square

3.4 Discrete-Continuous Case: $type(B) = \text{discrete}$, $type(A) = \text{continuous}$

The third case, when one class is continuous and the other is discrete, is done by extending the direct construction (basic or generalized) introduced above.

The idea is to use the product space $\mathbb{R} \times \mathbb{N}$ and the induced product of measures, i.e., the product of the discrete measure and the Lebesgue measure. We do that by constructing a continuous-to-continuous relation and then taking only points where the B coordinate is an integer.

Specifically, given a hybrid class diagram and a solution to the inequalities (*), we first obtain an instance $I_{\mathbb{R}}$ by the direct construction (basic or generalized). Based on $I_{\mathbb{R}}$, we define an instance I_{mix} by:

$$A^{I_{mix}} = A^{I_{\mathbb{R}}}, \quad B^{I_{mix}} = B^{I_{\mathbb{R}}} \cap \mathbb{N},$$

and

$$R^{I_{mix}} = R^{I_{\mathbb{R}}} \cap (\mathbb{R} \times \mathbb{N}).$$

Visually, this construction can be illustrated as taking the shaded area selected as R^I for the continuous case and considering its intersection with the vertical lines $\{1\} \times \mathbb{R}, \{2\} \times \mathbb{R}, \dots, \{b\} \times \mathbb{R}$. A formal proof of correctness follows the statement of the proposition.

Proposition 35 *For a hybrid class diagram as in Problem 1, where $type(A) = \text{continuous}$, $type(B) = \text{discrete}$, and a solution r, a, b of (*), the construction gives a legal instance that satisfies (**).*

Proof. We show first that cardinality constraints are satisfied for both sides of the associations. Let $\beta \in B^{I_{mix}}$. By the correctness of $I_{\mathbb{R}}$, we have that $\mu(R^{I_{mix}}(\beta)) = \mu(R^{I_{\mathbb{R}}}(\beta)) = r/b$ and by the arguments that proved Lemma 31, it means that the measure of the set associated with every element of $B^{I_{mix}}$ satisfies the cardinality constraints. For the other side, we use the fact that the number of integers contained in an interval $(x_1, x_2]$ is $\lfloor x_2 - x_1 \rfloor$ or $\lceil x_2 - x_1 \rceil$ (respectively, floor and ceiling of $x_2 - x_1$). Therefore, $\mu(R^{I_{mix}}(\alpha)) = |R^{I_{\mathbb{R}}}(\alpha) \cap \mathbb{N}| \in \{\lfloor r/a \rfloor, \lceil r/a \rceil\}$ which, by arguments similar to those used to prove Lemma 31, satisfies the cardinality constraints.

We now show that the instance satisfies (**). Clearly, the size of $A^{I_{mix}}$ is a , and the size of $B^{I_{mix}}$ is b . It remains to show that the size of $R^{I_{mix}}$ is r ,

as follows. According to the definition of $R^{I_{mix}}$, it follows that the number of intervals assigned to $A^{I_{mix}}$ members is

$$\sum_{\alpha \in A^{I_{mix}}} \int_{\beta \in R^{I_{mix}}(\alpha)} 1$$

which is equal to $a(r/b)$. On the other hand the number of integers associated to $B^{I_{mix}}$'s members is

$$\int_{\beta \in R(\alpha)} \sum_{\alpha \in A^{I_{mix}}} 1.$$

By Fubini theorem [1] it follows that these two expressions must yield the same value and, therefore, $\mu^2(R^{I_{mix}}) = r$. \square

3.5 One construction for Discrete and Continuous

Lastly, we present a unified construction for the discrete, continuous, an mixed case. This construction works only for some solutions of (*). We are not aware of a unified construction that works for all solutions of (*).

Specifically, when a and b are relatively prime, i.e, when $GCD(a, b) = 1$, the trick that we applied for the mixed discrete and continuous case works also for the discrete-discrete case. We say that this gives one construction that applies to all cases because the construction is essentially an adaption of the continuous-to-continuous to the two other cases.

For hybrid class diagram of Problem 1, where $type(A) = type(B) = \mathbf{discrete}$, and a solution r, a, b of (*) where $GCD(a, b) = 1$, consider the following construction: Get a legal instance $I_{\mathbb{R}}$ by the direct construction (basic or generalized) and define $I_{\mathbb{N}}$ by

$$A^{I_{\mathbb{N}}} = A^{I_{\mathbb{R}}} \cap \mathbb{N}, \quad B^{I_{\mathbb{N}}} = B^{I_{\mathbb{R}}} \cap \mathbb{N}$$

and

$$R^{I_{\mathbb{N}}} = R^{I_{\mathbb{R}}} \cap (\mathbb{N} \times \mathbb{N}).$$

Proposition 36 *This creates an instance that satisfies the cardinality constraints on the association in Problem 1.*

Proof. By the construction of $I_{\mathbb{R}}$, we have that, for all $\alpha \in A^{I_{\mathbb{R}}}$ and $\beta \in B^{I_{\mathbb{R}}}$, $R^{I_{\mathbb{R}}}(\alpha)$ and $R^{I_{\mathbb{R}}}(\beta)$ are semi-open intervals (in the modular space) of lengths r/b and r/a , respectively. In particular, by the argument used in the proof of the continuous-discrete case given above, the number of integers they may contain is given by $R^{I_{\mathbb{R}}}(\alpha) \cap \mathbb{N} \in \{\lfloor r/a \rfloor, \lceil r/a \rceil\}$ and $R^{I_{\mathbb{R}}}(\beta) \cap \mathbb{N} \in \{\lfloor r/b \rfloor, \lceil r/b \rceil\}$. It is not hard to see that every solution of (*), must satisfy $\{\lfloor r/b \rfloor, \lceil r/b \rceil\} \subseteq [m_A, x_A]$ and $\{\lfloor r/a \rfloor, \lceil r/a \rceil\} \subseteq [m_B, x_B]$. \square

Proposition 37 *If $GCD(a, b) = 1$, i.e., when a and b are relatively prime, the construction gives a legal instance that satisfies (**).*

Proof. Omitted due to space limitations. \square

4 Related Work

Many authors studied the existence of legal instances of models. In applications, this problem is equivalent to detecting whether the model is finitely satisfiable/satisfiable or not. In conceptual modeling, the main approach for detection of finite satisfiability problems involves reduction of finite satisfiability to solvability of a system of linear inequalities. The fundamental work is that of [2, 5], that results in a system of inequalities whose size is polynomial in the size of the diagram. Calvanese and Lenzerini, in [6], extend this method to diagrams with class hierarchy constraints. The size of the resulting system of inequalities is exponential in the size of the class diagram. Maraee and Balaban, in [3, 7] (www.cs.bgu.ac.il/~modeling), extend the method of [2] to apply to diagrams with class hierarchy constraints, generalization-set constraints, qualifier constraints, and association class constraints. The size of the resulting system is polynomial in the size of the diagram, but the method does not fully apply to class diagrams with complex multiple inheritance structure. The correlation between multiplicity constraints and linear programming is employed in several applications of class diagrams, e.g., in configuration management [8].

The idea of combining discrete and continuous mathematics for modeling is also studied in the context of dynamical systems and control [9]. Specifically, the, so called, hybrid automata are models where automata and differential equations are combined to allow for analysis of phenomena where discrete transitions and continuous state dynamics interact in ways that do not allow for analysis using neither pure discrete methods nor pure continuous methods [10]. The work presented in this paper stems from the same motivation but focuses on static models that describe the structure of a phenomena rather than its dynamics. In a future work, it might be interesting to study systems where both structure and dynamics are hybrid and examine how hybrid class diagrams combine with hybrid automata.

A mixed integer linear program is an optimization program involving continuous and integer variables. They are heavily used in practice for solving problems in transportation and manufacturing, airline crew scheduling, vehicle routing, production planning models, and more [11]. This paper lays foundations for seamless integration of conceptual and optimization modeling where the mathematical theory serves for bridging the conceptual modeling and mathematical programming because integration of software engineering modeling techniques with mathematical programming can be based on the generalized model of hybrid class diagrams.

Chenouard et al. present in [12] the s-COMMA, an extensible MDD platform for modeling CSPs. The authors present a visual modeling language using UML class diagram notation that combines the declarative aspects of constraint programming with the useful features of object-oriented languages. Using Meta-Model strategies, the authors present a transformation rules of the visual model to solver (executable) models. However, some of its features limits its use in engineering design, such as limited support for defining continuous CSPs. In [13], a *Model Based Mathematical Programming* framework is presented to provide a

designer with improved capabilities for modeling the *component sizing problems*. The authors present a formal capture of mathematical programming formalisms of GAMS domain using Meta-Models, representing GAMS compliant models in SysML using profiles and model transformations to automatically generate an executable representation in GAMS. These models may be simplified and made practical with the model of hybrid class diagrams presented here.

Choobineh [14] extended the Entity-Relationship model with notations for mathematical programming models. Lofo [15]) presented a specialization of UML class diagrams and UML statechart diagrams to facilitate the communication of models between engineers and domain experts. The visual languages are usually designed for human-to-human interaction and are not supported by formal mathematical semantics. Hybrid class diagrams can support this work by providing formal semantics for model transformations and other automated tools.

5 Future Work

Several directions remain for future work. Study the types of optimization problems, development of domain specific languages, study of their computational properties, and the construction of software engineering support for management of integrated optimization problems. Software engineering support may include a catalog of analysis, design, correctness and quality patterns, automated refactoring, reference models, and automated customization methods. A special challenge involves study of the gaps between conceptual modeling parameters and priorities to those of mathematical programming. An approach towards closing this gap can rely on model transformation techniques with formal semantics and proven properties. Future research may also include user experience study to examine productivity gains and acceptance parameters.

References

1. Royden, H.: Real Analysis. Prentice Hall; 3 edition (1988)
2. Lenzerini, M., Nobili, P.: On the Satisfiability of Dependency Constraints in Entity-Relationship Schemata. *Information Systems* **15** (1990) 453–461
3. Maraee, A., Makarenkov, V., Balaban, B.: Efficient recognition and detection of finite satisfiability problems in uml class diagrams: Handling constrained generalization sets, qualifiers and association class constraints. In: *MCCM*. (2008)
4. Feinerer, I., Salzer, G.: Consistency and minimality of uml class specifications with multiplicities and uniqueness constraints. In: *Proceedings of the 1st IEEE IFIP International Symposium on Theoretical Aspects of Software Engineering*, Shanghai, China. (2007)
5. Thalheim, B.: Fundamentals of cardinality constraints. In: *The 11th International Conference on the Entity-Relationship Approach*, Springer-Verlag (1992) 7–23
6. Calvanese, D., Lenzerini, M.: On the interaction between isa and cardinality constraints. In: *The 10th IEEE Int. Conf. on Data Engineering*. (1994)
7. Maraee, A.: Efficient methods for solving finite satisfiability problems in uml class diagrams. Master's thesis, Ben-Gurion University of the Negev (2007)

8. Falkner, A., Feinerer, I., Salzer, G., Schenner, G.: Computing product configurations via uml and integer linear programming. *International Journal of Mass Customisation* (2009) Accepted for publication.
9. Liberzon, D.: *Switching in Systems and Control* (Systems & Control: Foundations & Applications). Birkhuser Boston (2003)
10. Henzinger, T.: The theory of hybrid automata. In: *Logic in Computer Science, 1996. LICS '96. Proceedings., Eleventh Annual IEEE Symposium on.* (1996) 278–292
11. A. Wolsey, L., L. Nemhauser, G.: *Integer and Combinatorial Optimization*. Wiley-Interscience (1999)
12. Chenouard, R., Granvilliers, L., Soto, R.: Model-driven constraint programming. In: *Proceedings of the 10th international ACM SIGPLAN conference on Principles and practice of declarative programming, ACM* (2008) 236–246
13. A. Shah, A.: *Combining Mathematical Programming And SYSML For Component Sizing As Applied To Hydraulic Systems*. Master's thesis, Georgia Institute of Technology (2010)
14. Choobineh, J.: A diagramming technique for representation of linear models. *Omega* **19** (1991) 43–51
15. Lopes, L., Fourer, R.: Object oriented modeling of multistage stochastic linear programs. *Operations Research and Cyber-Infrastructure* (2009) 21–42