Electronic Communications of the EASST
Volume 41 (2011)

Proceedings of the Tenth International Workshop on Graph Transformation and Visual Modeling Techniques (GTVMT 2011)

Propagation of Constraints along Model Transformations Based on Triple Graph Grammars

Hartmut Ehrig , Frank Hermann , Hanna Schölzel , and Christoph Brandt

14 pages

Guest Editors: Fabio Gadducci, Leonardo Mariani
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Propagation of Constraints along Model Transformations Based on Triple Graph Grammars

Hartmut Ehrig [1], Frank Hermann [1], Hanna Schölzel [1], and Christoph Brandt [2]

[1] Institut für Softwaretechnik und Theoretische Informatik, Technische Universität Berlin,
[ehrig,frank,hannas]@cs.tu-berlin.de
[2] SECAN-Lab, Université du Luxembourg,
christoph.brandt@uni.lu,

Abstract: Model transformations based on triple graph grammars (TGGs) have been applied in several practical case studies and they convince by their intuitive and descriptive way of specifying bidirectional model transformations. Moreover, fundamental properties have been extensively studied including syntactical correctness, completeness, termination and functional behaviour. But up to now, it is an open problem how domain specific properties that are valid for a source model can be preserved along model transformations such that the transformed properties are valid for the derived target model. In this paper, we analyse in the framework of TGGs how to propagate constraints from a source model to an integrated and target model such that, whenever the source model satisfies the source constraint also the integrated and target model satisfy the corresponding integrated and target constraint. In our main new results we show under which conditions this is possible. The case study shows how this result is successfully applied for the propagation of security constraints in enterprise modelling between business and IT models.

Keywords: model transformation, graph constraints, security requirements, triple graph grammars

1 Introduction

Model integration and transformation between models as well as the compliance of such models with concrete security requirements have already been studied in different application domains, especially in the context of enterprise modelling [2]. In detail, it was possible to present how triple graph grammars (TGGs) in the sense of Schürr [14] can be used to realize the integration and transformation of those models. In addition to that, graph constraints [4] were utilized to verify that business and IT models comply with given security requirements.

However, it remained an open question how graph constraints valid for an IT model can be soundly propagated towards a corresponding business model. For example, the IT constraint on the left of Fig. 1 (public communication has to be encrypted) should be transformed into a corresponding business constraint (communication over public lines has to be filtered, right of Fig. 1). This problem was identified as an operational need in the decentralized organizational environment of Credit Suisse [2], where security requirements developed for IT models needed

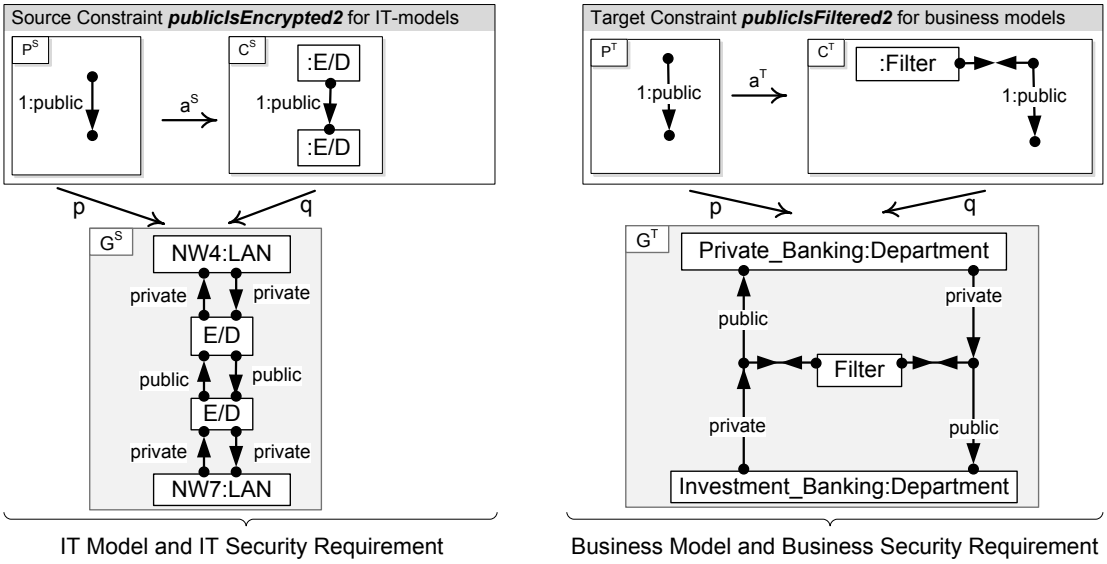


Figure 1: IT and business models with security requirements

to be understood from the point of view of the corresponding business models in order to ensure that the different persons responsible for the business models, IT models and security requirements will be able to integrate, transform and verify these models successfully. While this paper presents the case study in concrete syntax the presented techniques are based on the underlying typed attributed abstract syntax graphs [4].

Furthermore, if an IT model satisfies the source constraint the corresponding business model should satisfy the target constraint. In general, given a requirement for a source model specified by a graph constraint we would like to construct a corresponding requirement for the corresponding target model with the following satisfaction property: *Whenever a source model satisfies the given source graph constraint then the target model, defined by the model transformation, satisfies the corresponding target graph constraint.* In Fig. 1 the source model G^S satisfies the source graph constraint $PC(a^S : P^S \to C^S)$, because for each match $p : P^S \to G^S$ (occurrence of the premise graph) there is morphism $q : C^S \to G^S$ (occurrence of the conclusion graph) with $q \circ a^S = p$.

In this paper we show under which conditions we are able to define a propagation from source graph to target graph constraints such that this satisfaction property is valid. First of all it makes sense to require strong functional behaviour of the model transformation, which implies that we have for each source model a unique target model. Moreover this allows for each source graph constraint $PC(a^S : P^S \to C^S)$ with premise P^S, conclusion C^S and embedding morphism a^S to obtain a unique target graph constraint $PC(a^T : P^T \to C^T)$ by applying the model transformation to P^S and C^S leading to P^T and C^T. For this construction we require that P^S and C^S are source models, i.e. $P^S, C^S \in VL_S$, where VL_S is the source language of the model transformation $MT : VL_S \Rrightarrow VL_T$. In this case the source graph constraint $PC(a^S : P^S \to C^S)$ is called MT-consistent and leads to a MT-consistent target graph constraint $PC(a^T : P^T \to C^T)$. In Sec. 2 we review model transformation based on triple graph grammars [5, 14, 15] and prepare our case study based on a model transformation from business to IT-models.

Our first main result in Sec. 3 shows that the satisfaction property stated above for the propaga-

tion of security constraints is valid for MT-consistent source and target constraints. In Sec. 4 we discuss how to extend the theory to the case of partially MT-consistent constraints where premise or conclusion consist only of model fragments, s.t. model transformations are not directly applicable. Our constructions and results are illustrated by a case study of security constraints in enterprise modelling. For the main results we only give proof ideas. For detailed proofs see [6, 13].

2 Model Transformation between Business and IT Models

Triple graph grammars (TGGs) [14] are a well known approach for bidirectional model transformations and we apply TGGs to define the model transformation of our case study between business and IT models. For this purpose we review main constructions and results of model transformations based on triple graph grammars [15, 5] in this section.

Integrated models are defined as pairs of source and target graphs, which are connected via a correspondence graph together with relating morphisms between these graphs. More precisely, a triple graph $G = (G^S \xleftarrow{s_G} G^C \xrightarrow{t_G} G^T)$ consists of three graphs G^S, G^C, and G^T, called source, correspondence, and target graphs, together with two graph morphisms $s_G : G^C \to G^S$ and $t_G : G^C \to G^T$.

A triple graph morphism $m : G \to H$ with $m = (m^S, m^C, m^T)$ consists of three graph morphisms $m^S : G^S \to H^S$, $m^C : G^C \to H^C$ and $m^T : G^T \to H^T$ such that $m^S \circ s_G = s_H \circ m^C$ and $m^T \circ t_G = t_H \circ m^C$. A typed triple graph G is typed over a triple graph TG by a triple graph morphism $type_G : G \to TG$ and a typed triple graph morphism $m : (G, type_G) \to (H, type_H)$ preserves the typing, i.e. $type_H \circ m = type_G$. Triple graphs may also contain attributed nodes and edges according to [5] and they form an $\mathcal{M}$-adhesive as well as weak adhesive HLR category for which several important formal results have been shown in [4].

$$\begin{array}{ccccc} G = (G^S & \xleftarrow{s_G} & G^C & \xrightarrow{t_G} & G^T) \\ {\scriptstyle m}\downarrow & {\scriptstyle m^S}\downarrow & {\scriptstyle m^C}\downarrow & & {\scriptstyle m^T}\downarrow \\ H = (H^S & \xleftarrow[s_H]{} & H^C & \xrightarrow[t_H]{} & H^T) \end{array}$$

Triple rules synchronously build up source and target graphs as well as their correspondence graphs, i.e. they are non-deleting. A triple rule tr is an injective triple graph morphism $tr = (tr^S, tr^C, tr^T) : L \to R$ and w.l.o.g. we assume tr to be an inclusion. Given an (almost) injective triple graph morphism $m : L \to G$, a triple graph transformation (TGT) step $G \overset{tr,m}{\Longrightarrow} H$ from G to a triple graph H is given by a pushout of triple graphs with comatch $n : R \to H$ and transformation inclusion $t : G \hookrightarrow H$. Given a sequence of TGT-steps $G_0 \overset{tr_1,m_1}{\Longrightarrow} G_1 \ldots \overset{tr_k,m_k}{\Longrightarrow} G_k$ its trace is given by $trace = t_k \circ \ldots t_2 \circ t_1$. A grammar $TGG = (TG, S, TR)$ consists of a triple type graph TG, a triple start graph S and a set TR of triple rules.

$$\begin{array}{ccccc} L = (L^S & \xleftarrow{s_L} & L^C & \xrightarrow{t_L} & L^T) \\ {\scriptstyle tr}\downarrow & {\scriptstyle tr^S}\downarrow & {\scriptstyle tr^C}\downarrow & & {\scriptstyle tr^T}\downarrow \\ R = (R^S & \xleftarrow[s_R]{} & R^C & \xrightarrow[t_R]{} & R^T) \end{array} \qquad \begin{array}{ccc} L & \xrightarrow{tr} & R \\ {\scriptstyle m}\downarrow & (PO) & \downarrow{\scriptstyle n} \\ G & \xrightarrow[t]{} & H \end{array}$$

Example 1 (Triple Rules) *The triple rules in Fig. 2 are part of the rules of the grammar TGG in [2]. They are presented in short notation, i.e. left and right hand sides of a rule are depicted in one triple graph. Elements, which are created by the rule, are labeled with green "++" and marked by green line colouring. The rule LANToDepartment creates LAN element in the IT model and a corresponding Department element in the Business model. The rule PublicToP-*

ublic generates public edges with gluing nodes in both domains simultaneously. The encryption/decryption nodes (E/D) are created in front and at the end of a public Reo connector (depicted as black arrows) in the rule EDToFilter, where in the Business model a Filter is attached to the corresponding public Reo connector.

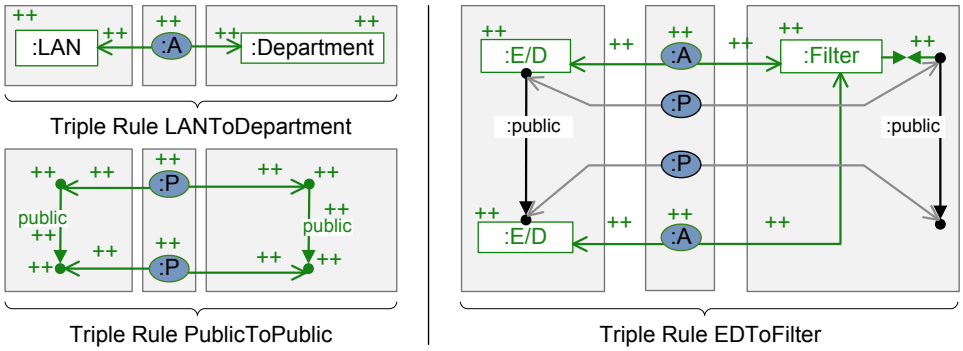
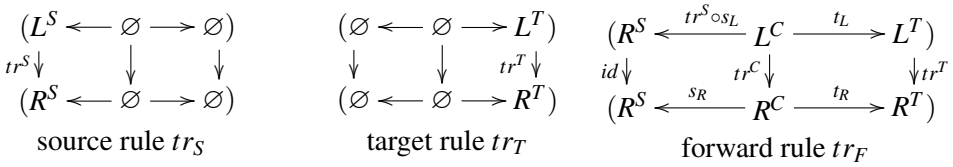


Figure 2: Some triple rules of the model transformation



The operational rules for model transformations based on TGGs are automatically derived from the set of triple rules TR [5]. From each triple rule tr we derive a forward rule tr_F for forward transformation sequences and a source rule tr_S for the construction resp. parsing of a model of the source language. By TR_S and TR_F we denote the sets of all source and forward rules derived from TR. The sets of backward rules TR_B and target rules TR_T are derived analogously as presented in [2].

Example 2 (Forward Rule) *The rule in Fig. 3 is the derived forward rule of the triple rule "EDToFilter" shown in Fig. 2. No new elements are added in the source graph by this rule. The "E/D" elements that are added by the triple rule have already to be present in a model to make this rule applicable. Whereas the corresponding and target graphs and the morphisms remain the same as in the triple rule.*

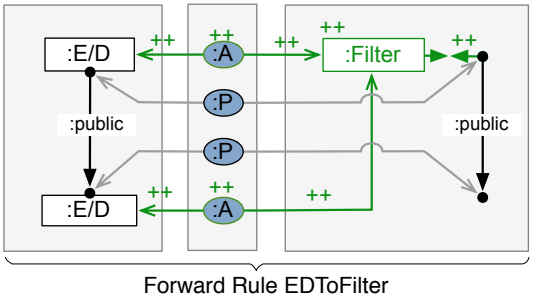


Figure 3: A derived forward rule

A set of triple rules TR and the start graph $\varnothing$ typed over a triple graph TG generate a visual language VL of integrated models, i.e. models with elements in the source, target and correspondence component. The source language VL_S and target language VL_T are derived by projection to the triple components, i.e. $VL_S = proj_S(VL)$ and $VL_T = proj_T(VL)$. For the S-component TG^S and T-component TG^T we denote by $VL(TG^S)$ and $VL(TG^T)$ the visual language of all source

and target models typed over TG^S and TG^T respectively. This means we have $VL_S \subseteq VL(TG^S)$ and $VL_T \subseteq VL(TG^T)$.

As presented in [5] the derived operational rules provide the basis to define model transformations based on source consistent forward transformations $G_0 \Rightarrow^* G_n$ via $(tr_{1,F}, \ldots, tr_{n,F})$, short $G_0 \overset{tr_F^*}{\Longrightarrow} G_n$. A forward sequence $G_0 \overset{tr_F^*}{\Longrightarrow} G_n$ is source consistent, if there is a source sequence $\varnothing \overset{tr_S^*}{\Longrightarrow} G_0$ such that the sequence $\varnothing \overset{tr_S^*}{\Longrightarrow} G_0 \overset{tr_F^*}{\Longrightarrow} G_n$ is match consistent, i.e. the S-component of each match $m_{i,F}$ of $tr_{i,F} (i = 1 \ldots n)$ is uniquely determined by the comatch $n_{i,S}$ of $tr_{i,S}$, where $tr_{i,S}$ and $tr_{i,F}$ are source and forward rules of the same triple rules tr_i. Thus, source consistency is a control condition for the construction of the forward sequence.

Definition 1 (Model Transformation Based on Forward Rules) A *model transformation sequence* is given by a tuple $(G_S, G_0 \overset{tr_F^*}{\Longrightarrow} G_n, G_T)$ consisting of a source graph G_S, an integrated graph $G = G_n$, and a target graph G_T, and a source consistent forward sequence $G_0 \overset{tr_F^*}{\Longrightarrow} G_n$ with $G_S = G_0^S$ and $G_T = G_n^T$. A *model transformation* $MT : VL(TG^S) \Rrightarrow VL(TG^T)$ based on forward rules is given by a set of model transformation sequences $(G_S, G_0 \overset{tr_F^*}{\Longrightarrow} G_n, G_T)$ with $G_S \in VL(TG^S)$ and $G_T \in VL(TG^T)$.

Model transformations based on forward rules using the control condition "source consistency" are syntactically correct and complete as shown in [5, 11]. *Correctness* means that for each source model G^S that is transformed into a target model G^T there is an integrated model $G = (G^S \leftarrow G^C \rightarrow G^T)$ in the language of integrated models VL generated by the TGG. *Completeness* ensures that for each valid source model there is always a forward transformation sequence that transforms it into a valid target model, and if the source model is not in VL_S then there is no source consistent forward transformation sequence. Therefore, we can apply the on-the-fly construction presented in [5] to any given source model and ensure that we derive a correct corresponding target model and if all source rules are creating then we can always ensure termination.

Example 3 (Model Transformation) *The model transformation from IT to business models via the forward rules of the triple rules in Ex. 1 transforms the source model G^S in Fig. 1 into the target model G^T in Fig. 1 as presented in [2]. Using backward and target rules we obtain a model transformation from business to IT models and all together we obtain a bidirectional model transformation, where both directions are useful in different phases of enterprise modelling. Especially a propagation of constraints from IT to business models will be considered in Sec. 3.*

3 Propagation of MT-consistent Constraints

The propagation of constraints along a given model transformation aims at translating requirements from the source domain to the corresponding target domain in order to verify them at corresponding target models. In this section we present a constructive approach for the propagation of constraints based on a given model transformation and we show by Thm. 1 how each source constraint can be propagated to an integrated and a target constraint for the languages of integrated and target models, respectively. In our main result Thm. 2 we show that under

suitable conditions the propagation preserves the validity of constraints. An (atomic) constraint $PC(a : P \to C)$ for a triple graph is given by a premise P and a conclusion C connected by a morphism $a : P \to C$. A graph G satisfies $PC(a : P \to C)$, if for each injective $p : P \to G$ there is an injective $q : C \to G$ with $q \circ a = p$. Atomic constraints can be combined to general constraints as usual by boolean operators.

The first important property for ensuring the creation of propagated constraints for a given source constraint is MT-consistency meaning that the constraint has to be compatible with the model transformation MT. We take into account a domain specific source language $\mathcal{L}_S$ which should be a sublanguage of VL_S, i.e. $\mathcal{L}_S \subseteq VL_S$.

Definition 2 (MT-consistent Constraints) Given a model transformation $MT : VL_S \Rrightarrow VL_T$, then a constraint is MT-consistent, if the corresponding condition below is satisfied.

constraint kind		typed over	condition
source constraint	$PC(a^S : P^S \to C^S)$	TG^S	$P^S, C^S \in \mathcal{L}_S \subseteq VL_S$
integrated constraint	$PC(a : P \to C)$	TG	$P, C \in VL$
target constraint	$PC(a^T : P^T \to C^T)$	TG^T	$P^T, C^T \in VL_T$

Moreover, the sound propagation of constraints is based on the notion of propagation consistency, which requires strong functional behaviour of the model transformation and one further technical condition.

Definition 3 (Propagation Consistency and Strong Functional Behaviour of Model Transformations) A model transformation MT is *propagation consistent*, if:

- The model transformation MT has *strong functional behaviour* with respect to $\mathcal{L}_S$. This means that matches are injective and for each source graph $G_S \in \mathcal{L}_S \subseteq VL_S$ the execution of MT terminates resulting in a model transformation sequence $(G_S, G_0 \overset{tr_F^*}{\Longrightarrow} G_n, G_T)$, and moreover, any two source consistent forward sequences $G_0 \overset{tr_F^*}{\Longrightarrow} G_n$ and $G_0 \overset{\overline{tr}_F^*}{\Longrightarrow} \overline{G}_m$ constructed via MT that cannot be extended by any further step via MT are switch-equivalent up to isomorphism, i.e. the rules of tr_F^* are a permutation of those in $\overline{tr}_F^*$, $n = m$ and $G_n \cong G_m$.
- Furthermore, each triple graph $G = (G^S \xleftarrow{s_G} G^C \xrightarrow{t_G} G^T) \in VL$ has to be left-linear, i.e. we have that s_G is injective.

Remark 1 (Checking Propagation Consistency) *Concerning the first condition (strong functional behaviour), we have presented in [11] how model transformations based on forward rules are checked for strong functional behaviour using the tool AGG for critical pair analysis. For the second condition (left linearity) it suffices to show that no rule is capable to transform a triple graph $G = (G^S \xleftarrow{s_G} G^C \xrightarrow{t_G} G^T)$ into a triple graph $H = (H^S \xleftarrow{s_H} H^C \xrightarrow{t_H} H^T)$ with non-injective s_H. This condition is ensured if there is no triple rule which simultaneously creates a correspondence element c and relates it to an existing source element $s \in L^S$, i.e. we require for all rules that $[c \in R^C \setminus L^C \wedge s_R(c) = s] \Rightarrow [s \notin L^S]$. Both conditions have been verified for our case study.*

In our first main result we show how each source constraint can be propagated into an inte-

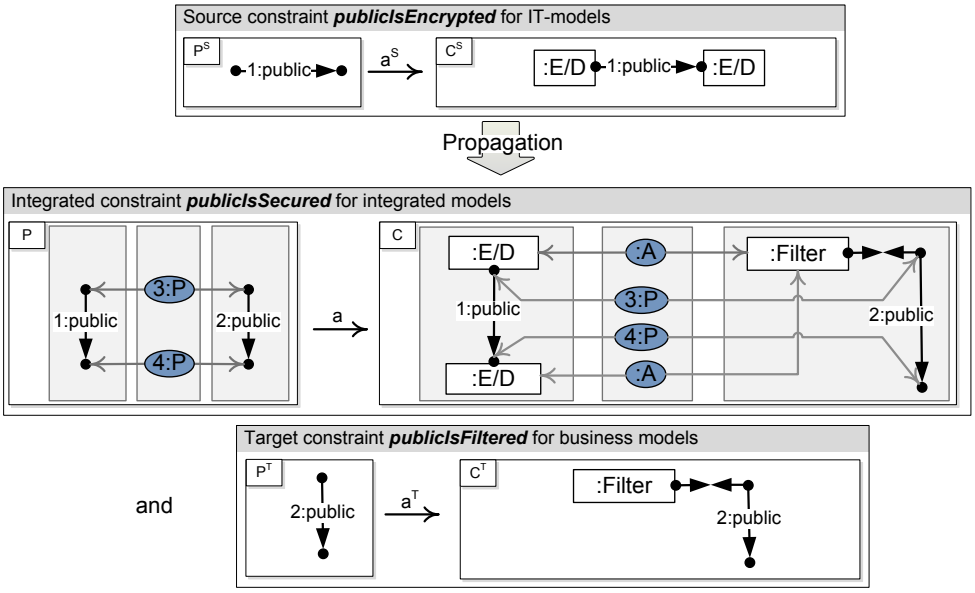


Figure 4: Source constraint as well as propagated integrated and target constraints

grated and a target constraint as shown by the example in Fig 4 which happens to be exactly the rule *EDToFilter*. Note that in general those graphs are not necessarily the same. Intuitively, the transformation steps of the premise graph P, which is contained in the conclusion graph C, are transferred to the transformation steps for C and the transformation of C is completed for the remaining parts in C.

Theorem 1 (Propagation and Restriction of Constraints) *Given a TGG model transformation MT with strong functional behaviour, then*

1. *an MT-consistent source constraint $PC(a^S)$ generates an MT-consistent integrated constraint $PC(a)$, called* propagated integrated constraint*, where a^S is the source component of a and diagrams (1) and (2) below commute using the trace morphisms of the transformation sequences.*

$$\begin{array}{ccccc} (P^S \leftarrow \emptyset \rightarrow \emptyset) = P_0 & \overset{tr_F^*}{\underset{trace_P}{\rightrightarrows}} & P_l & = P & \\ \downarrow{\scriptstyle (a^S,\emptyset,\emptyset)} & (1) & \downarrow{\scriptstyle a_l} & \searrow{\scriptstyle a} & \\ (C^S \leftarrow \emptyset \rightarrow \emptyset) = C_0 & \overset{tr_F^*}{\underset{trace_{C1}}{\rightrightarrows}} & C_l & \overset{tr'^*_F}{\underset{trace_{C2}}{\rightrightarrows}} & C_n = C \end{array} \quad (2)$$

2. *an MT-consistent integrated constraint $PC(a)$ can be restricted to and an MT-consistent target constraint $PC(a^T : P^T \rightarrow C^T)$.*

Proof Idea. 1. Given an MT-consistent $PC(a^S : P^S \rightarrow C^S)$ we have $P^S, C^S \in \mathcal{L}_S \subseteq VL_S$. Now the completeness of model transformations shown in [5, 11] implies $P_0 \overset{tr_F^*}{\Longrightarrow} P_l = P$ with $P_0 = (P^S \leftarrow \emptyset \rightarrow \emptyset)$ and $P \in VL$. This model transformation sequence can be extended firstly along $(a^S, \emptyset, \emptyset) : P_0 \rightarrow C_0$ leading to $C_0 \overset{tr_F^*}{\Longrightarrow} C_l$ with (1) commutative and secondly

by strong functional behaviour of MT to $C_0 \overset{tr_F^*}{\Longrightarrow} C_l \overset{tr_F'^*}{\Longrightarrow} C_n$ with $C_n = C \in VL$. This implies that $a : P \rightarrow C$ defined by triangle (2) is MT-consistent, i.e. $P, C \in VL$.

2. Given an MT-consistent integrated constraint $PC(a : P \rightarrow C)$ we have $P, C \in VL$. By definition of $VL_T = proj_T(VL)$ we have $P^T, C^T \in VL_T$ and hence the required MT-consistency of the target constraints. □

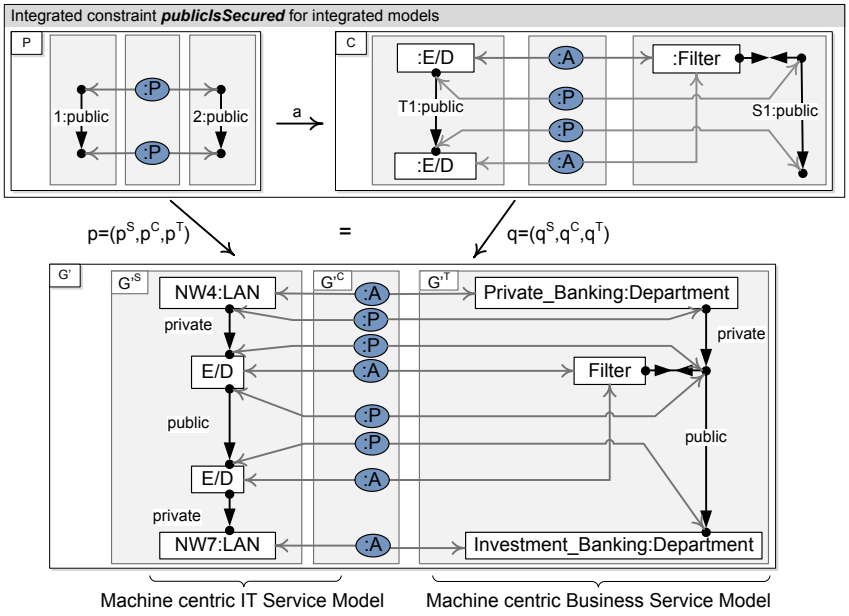


Figure 5: Propagated Integrated Constraint

Example 4 (Propagation) *According to Rem. 1 the model transformation of our case study is propagation consistent. Furthermore, the source constraint is MT-consistent, such that we can apply Thm. 1 and derive the propagated integrated and target constraints in Fig. 5. For better visibility we take a subgraph $G' \subseteq G$ of the integrated model G in Ex. 3, such that the source model G'^S is similarly transformed into the target model G'^T. Now, the integrated model G' satisfies the propagated integrated constraint, i.e. for any injective occurrence $p : P \rightarrow G'$ of the integrated premise P in G' there is an injective occurrence $q : C \rightarrow G'$ compatible with the constraint morphism a, i.e. $p = q \circ a$.*

In order to generally ensure the validity of propagated constraints we provide a suitable static condition on constraints by Def. 4 below that can be checked automatically (see Remark 2). Essentially, the condition requires that whenever the premise graph P of a constraint is found in an integrated model, then its occurrence is already fully determined by the source and correspondence component. This condition is not very restrictive, because it must only hold for premise graphs of constraints and not for all integrated graphs in VL.

Definition 4 (Admissable Premise) Given a TGG with triple language VL. The premise P of an integrated constraint $PC(a : P \rightarrow C)$ is called *admissable*, if for any injective morphisms $p, p' : P \rightarrow G$ with $G \in VL$ we have $p^S = p'^S$ and $p^C = p'^C$ implies $p^T = p'^T$.

Remark 2 (Checking Admissability) *It suffices to show that the internal morphism $t_P : P^C \to P^T$ of the premise graph P is surjective on nodes, and furthermore, for each edge type occurring in the target component P^T of P we have that there are no parallel edges in any triple graph $G \in VL$ of this type. The latter can be verified - especially in our case study - by checking that the triple rules do not create edges of those types separately, but always together with an adjacent node.*

The premise graph of a propagated target constraint may occur with a target model at places that do not correspond to occurrences of the premise graph of the source constraint in the source model, because differently typed source elements may be transformed into target elements of the same type. From the application point of view, it is clear that the preservation of properties of the source model can be ensured only at corresponding occurrences in the target model. Theorem 2 below shows that the validity of source constraints is preserved at those places using the notion of weak satisfaction. More precisely, given an integrated constraint $PC(a : P \to C)$, then a target model G^T weakly satisfies a target constraint – written $(G^T \models^w PC(a^T))$ – if for all injective $p^T : P^T \to G^T$ which can be extended to an injective integrated morphism $p : P \to G$ there exists an injective $q^T : C^T \to G^T$ with $q^T \circ a^T = p^T$. Furthermore, the theorem shows that the validity is completely preserved for the propagated integrated constraint. It is formulated for (atomic) constraints $PC(a^S : P^S \to C^S)$ but can be extended to general constraints.

Theorem 2 (Validity of Propagation for MT-Consistent Constraints) *Given a propagation consistent model transformation MT acc. to Def. 3, and given an MT-consistent source graph constraint $PC(a^S : P^S \to C^S)$ with an MT-consistent propagated integrated constraint $PC(a : P \to C)$ and a propagated target constraint $PC(a^T : P^T \to C^T)$ according to Thm. 1, such that P is admissable, then we have for all $G^S \in \mathcal{L}_S$ with model transformation sequence $(G^S, G_0 \overset{tr_F^*}{\Longrightarrow} G, G^T)$:*

$$\left(G^S \models PC(a^S) \right) \Rightarrow \left(G \models PC(a) \wedge G^T \models^w PC(a^T) \right).$$

This means that given the source graph satisfies the source constraint, then the integrated graph satisfies the propagated integrated constraint and the target graph weakly satisfies the propagated target constraint.

Proof Idea. The main idea is to show that $G^S \models PC(a^S : P^S \to C^S)$ implies $G \models PC(a : P \to C)$ where both are related by diagrams (1) and (2) of Thm. 1. Similar to the proof of Thm 1 we are firstly able to extend $C_0 \overset{tr_F^*}{\Longrightarrow} C_l \overset{tr_F'^*}{\Longrightarrow} C_n = C$ along $(q^S, \emptyset, \emptyset) : C_0 \to G_0$ with $G_0 = (G^S \leftarrow \emptyset \to \emptyset)$ to $G_0 \overset{tr_F^*}{\Longrightarrow} G_l \overset{tr_F'^*}{\Longrightarrow} G_n$ with $q_n : C_n \to G_n$ and secondly by strong functional behaviour to $G_0 \overset{tr_F^*}{\Longrightarrow} G_l \overset{tr_F'^*}{\Longrightarrow} G_n \overset{tr_F''^*}{\Longrightarrow} G_m = G$ with $G \in VL$. Using $q = trace_G \circ q_n : C \to G$ where $trace_G = trace(G_n \overset{tr_F''^*}{\Longrightarrow} G_m)$ we can deduce $q \circ a = p$ using left linearity (see Def. 3) and admissability of premise P (see Def. 4). □

Example 5 (Validity of Propagated Constraints) *The premise graph of the integrated constraint in Fig. 5 is admissable, which we verified via Remark 2. According to Ex. 4 the model transformation is propagation consistent and the source as well as the propagated constraints are MT-consistent. Therefore, we can apply Thm. 2 for showing that the integrated model G' satisfies the propagated integrated constraint and the target model G'^T weakly satisfies the propagated target constraint $PC(a^T : P^T \to C^T)$ in Fig. 5, i.e. the constraint holds at all structures that correspond*

to occurrences of the source constraint in the source model.

4 Propagation of Partially MT-consistent Constraints

In this section we discuss how to generalize constructions and results of Sec. 3 to the case of partially MT-consistent constraints, i.e. for the source constraint $PC(a^S : P^S \to C^S)$ we may have $P^S \notin \mathcal{L}_S \subseteq VL_S$ or $C^S \notin \mathcal{L}_S$ as shown in Fig. 6a. Note that in Fig. 6a black bullets are missing s.t. P^S is a model fragment in contrast to P^S in Fig. 4. Moreover, P^S in concrete syntax in Fig. 6a corresponds to a single node in abstract syntax in Fig. 6b. If $P^S \notin \mathcal{L}_S$ or $C^S \notin \mathcal{L}_S$ they can be considered as model fragments and we show now how to handle this important more general case. The main idea is to use an extended model transformation approach for model fragments introduced in [13] based on the general framework of graph transformation with borrowed context (BC) [8]. Intuitively, BC-transformations allow for partial matching of the forward rules in the S-component. The missing context that is required by the left hand side L_F of the forward rule is borrowed and the instance graph is extended by this context. More precisely each BC-forward transformation step consists of two POs (1) and (2), where in PO (1) the partial match is completed and in PO (2) we have a forward transformation step with total match.

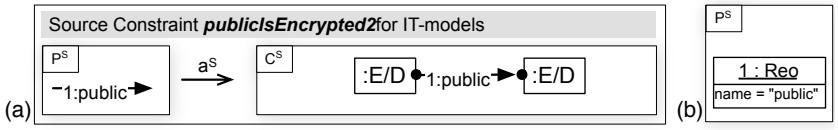


Figure 6: Partially MT-consistent source constraint (a) and abstract syntax of P^S (b)

Definition 5 (BC-Forward and BC-Model Transformation) Given triple rules TR with corresponding forward rules TR_F then

1. A *BC-forward transformation* $\widehat{G}_0 \overset{tr_F^*}{\Longrightarrow}_{BC} \widehat{G}_n$ via TR_F is given by BC-forward transformation steps $\widehat{G}_{i-1} \overset{tr_{i,F},\widehat{m}_{i,F},d_i}{\Longrightarrow} \widehat{G}_i$ for $i = 1,\ldots,n$ consisting of POs (1) and (2) where in PO (1) the partial match $m' : L_{i,F} \rightharpoonup \widehat{G}_{i-1}$ – given by injective morphisms $\widehat{m}_{i,F}$ and d_i – is extended to a total injective match $m^+_{i,F} : L_{i,F} \to \widehat{G}^+_{i-1}$ and in PO(2) we have a forward transformation step with total injective match as in Sec. 2. Moreover we require that the C- and T-components d_i^C and d_i^T of d_i are identities. Note that the S-component $tr^S_{i,F}$ is the identity of R_i^S by construction of TR_F, i.e. $tr^S_{i,F} = id_{R_i^S}$.

$$\begin{array}{ccc} D_i & \xrightarrow{d_i} & L_{i,F} \\ {\scriptstyle \widehat{m}_{i,F}}\downarrow & \swarrow_{m'} & \\ \widehat{G}_{i-1} & & \end{array} \qquad \begin{array}{ccccc} D_i & \xrightarrow{d_i} & L_{i,F} & \xrightarrow{tr_{i,F}} & R_{i,F} \\ {\scriptstyle \widehat{m}_{i,F}}\downarrow & (1) & \downarrow{\scriptstyle m^+_{i,F}} & (2) & \downarrow{\scriptstyle n_{i,F}} \\ \widehat{G}_{i-1} & \xrightarrow[h_i]{} & \widehat{G}^+_{i-1} & \xrightarrow[h^+_i]{} & \widehat{G}_i \end{array}$$

2. A *BC-model transformation sequence* $(G_S, \widehat{G}_0 \overset{tr_F^*}{\Longrightarrow}_{BC} \widehat{G}_n, G_T)$ consists of a source consistent BC-forward transformation $\widehat{G}_0 \overset{tr_F^*}{\Longrightarrow}_{BC} \widehat{G}_n$ via TR_F with source model $G_S = \widehat{G}^S_n$ and target model $G_T = \widehat{G}^T_n$. A *BC-model transformation* $MT_{BC} : VL(TG^S) \Rrightarrow_{BC} VL(TG^T)$ con-

sists of BC-model transformation sequences $(G_S, \widehat{G}_0 \overset{tr_F^*}{\Longrightarrow}_{BC} \widehat{G}_n, G_T)$ with $G_S \in VL(TG^S)$ and $G_T \in VL(TG^T)$.

Remark 3 *Source consistency of BC-forward transformations is based on partial BC-match consistency [13], where both notions are defined in analogy to source and match consistency in the standard case without BC [5].*

BC-forward and BC-model transformations can be extended by Fact 1 (with proof in [13]) to forward and model transformations in the sense of Sec. 2.

Fact 1 (Extension of BC-Forward and BC-Model Transformations) *Given a BC-forward transformation sequence $\widehat{G}_0 \overset{tr_F^*}{\Longrightarrow}_{BC} \widehat{G}_n$ with $\widehat{G}_0 = (\widehat{G}_0^S \leftarrow \emptyset \rightarrow \emptyset)$ there is an extension to a forward transformation sequence $G_0 \overset{tr_F^*}{\Longrightarrow} G_n$ with $G_0 = (\widehat{G}_n^S \leftarrow \emptyset \rightarrow \emptyset)$ and $G_n = \widehat{G}_n$.*
Moreover, each BC-model transformation sequence $(G_S, \widehat{G}_0 \overset{tr_F^*}{\Longrightarrow}_{BC} \widehat{G}_n, G_T)$ with $G_S = \widehat{G}_n^S$, $G_T = \widehat{G}_n^T$ can be extended to a model transformation sequence $(G_S, G_0 \overset{tr_F^*}{\Longrightarrow} G_n, G_T)$ with the same G_S, G_T satisfying $G_S = G_0^S$, $G_T = G_n^T$ and $G_n = \widehat{G}_n$.

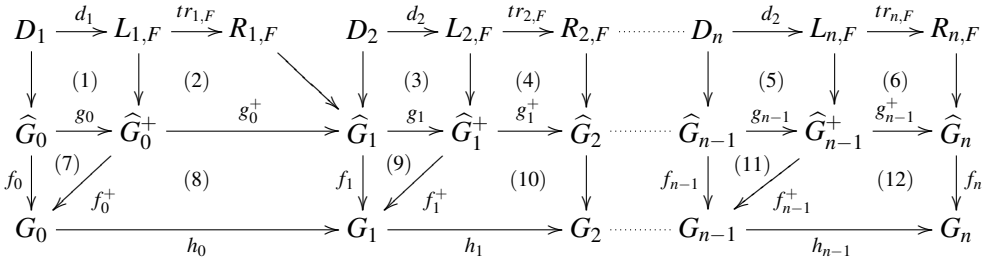


Figure 7: Extension of a BC-forward Transformation Sequence

Now correctness of model transformation shown in [5, 11] can be extended by Fact 2 (with proof in [13]) to BC-model transformation.

Fact 2 (Correctness of BC-Model Transformations) *Each BC-model transformation is correct, i.e. for all BC-model transformation sequences $(G_S, \widehat{G}_0 \overset{tr_F^*}{\Longrightarrow}_{BC} \widehat{G}_n, G_T)$ we have $\widehat{G}_n \in VL$ with $\widehat{G}_0^S \subseteq \widehat{G}_n^S = G_S \in VL_S$ and $\widehat{G}_n^T = G_T \in VL_T$.*

Finally we show how to propagate a partially MT-consistent source constraint (see Fig. 6) to MT-consistent target constraint as shown already in Fig. 4 using the BC-model transformations in Def. 6. Intuitively a partially MT-consistent source constraint $PC(a^S : P^S \rightarrow C^S)$ is a source constraint such that we have a BC-model transformation sequence from $(P^S \leftarrow \emptyset \rightarrow \emptyset)$ to $\widehat{P}$ and from $(C^S \leftarrow \emptyset \rightarrow \emptyset)$ to $\widehat{C}$ leading to a propagated integrated constraint $PC(\widehat{a} : \widehat{P} \rightarrow \widehat{C})$ together with a corresponding propagated source constraint $PC(\widehat{a}^S : \widehat{P}^S \rightarrow \widehat{C}^S)$ and target constraint $PC(\widehat{a}^T : \widehat{P}^T \rightarrow \widehat{C}^T)$

Definition 6 (Partially MT-Consistent Source Constraint) A source constraint $PC(a^S)$ with $a^S : P^S \rightarrow C^S$ is called *partially MT-consistent* if there exist BC-model transformation sequences

$\widehat{P}_0 \overset{tr_F^*}{\Longrightarrow}_{BC} \widehat{P}_l, \widehat{C}_0 \overset{tr_F^*}{\Longrightarrow}_{BC} \widehat{C}_l \overset{tr_F^*}{\Longrightarrow}_{BC} \widehat{C}_n$ as shown below leading to a morphism $\widehat{a} = trace(tr_F'^*) \circ \widehat{a}_l$, such that the diagram below commutes and we have $\widehat{P}^S, \widehat{C}^S \in \mathcal{L}_S$:

$$\begin{array}{ccccc}
(P^S \leftarrow \emptyset \rightarrow \emptyset) = \widehat{P}_0 & & \overset{tr_F^*}{\Longrightarrow}_{BC} & & P_l \quad = \widehat{P} \\
\downarrow{\scriptstyle (a^S,\emptyset,\emptyset)} & & & & \downarrow{\scriptstyle \widehat{a}_l} \quad \searrow{\scriptstyle \widehat{a}} \\
(C^S \leftarrow \emptyset \rightarrow \emptyset) = \widehat{C}_0 & & \overset{tr_F^*}{\Longrightarrow}_{BC} & & \widehat{C}_l \overset{tr_F'^*}{\Longrightarrow}_{BC} \widehat{C}_n \quad = \widehat{C}
\end{array}$$

Moreover, $PC(\widehat{a} : \widehat{P} \rightarrow \widehat{C})$ is called *propagated integrated* constraint, $PC(\widehat{a}^S : \widehat{P}^S \rightarrow \widehat{C}^S)$ *propagated source* constraint, and $PC(\widehat{a}^T : \widehat{P}^T \rightarrow \widehat{C}^T)$ *propagated target* constraint, where $\widehat{a}^S$ and $\widehat{a}^T$ are the source and target components of $\widehat{a}$.

This allows us to present our second main result – the validity of propagation of partially MT-consistent source constraints – as a generalisation of Thm. 2.

Theorem 3 (Propagation of Partially *MT*-consistent Source Constraints) *Given a propagation consistent model transformation MT and a partially MT-consistent source constraint* $PC(a^S : P^S \rightarrow C^S)$*, then we have MT-consistent propagated constraints* $PC(\widehat{a} : \widehat{P} \rightarrow \widehat{C})$ *with* $PC(\widehat{a}^S)$ *and* $PC(\widehat{a}^T)$*. If* $\widehat{P}$ *is admissable, then we have for all* $G^S \in \mathcal{L}_S$ *with model transformation sequence* $(G^S, G_0 \overset{tr_F^*}{\Longrightarrow} G, G^T)$*:*

$$\left(G^S \models PC(\widehat{a}^S) \right) \Rightarrow \left(G \models PC(\widehat{a}) \wedge G^T \models^w PC(\widehat{a}^T) \right).$$

Proof Idea. MT-consistency of $PC(\widehat{a})$, $PC(\widehat{a}^S)$, and $PC(\widehat{a}^T)$ follows from correctness of BC-model transformations (see Fact 2) and $\widehat{P}^S, \widehat{C}^S \in \mathcal{L}_S$ by assumption. This allows us to apply Thm. 2. □

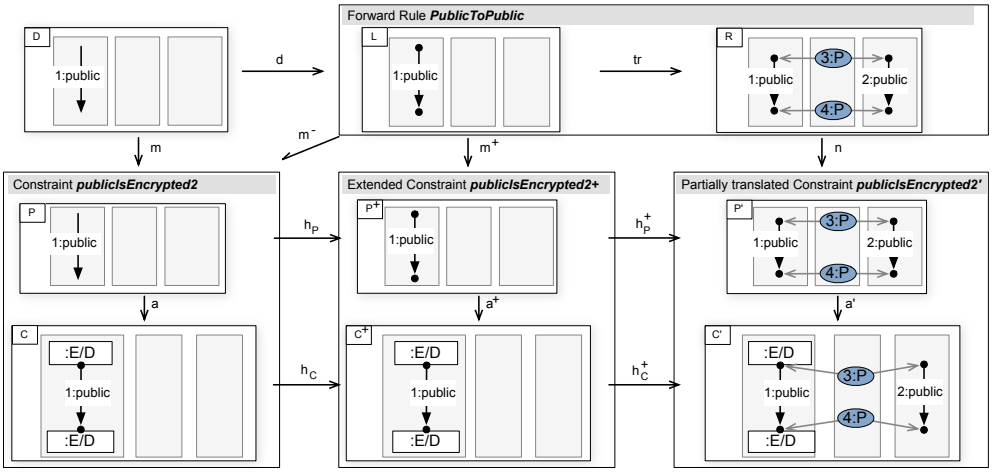


Figure 8: Source constraint and propagated constraint

Example 6 (Propagation of a Partially *MT*-consistent Source Constraint) *The constraint publicIsEncrypted2 in Fig. 8 is a partial model, as shown in Fig. 6. Therefore there is no total match from the left-hand-side of the rule L to P, but only a partial match* m^-*. This leads to the injective*

span $(L \xleftarrow{d} D \xrightarrow{m} G)$ *with D being the domain of* m^-. *With a pushout over D the black bullet nodes are "borrowed" from L and a standard triple graph transformation over the rule can be performed. After this step the premise graph is already completely translated and the conclusion graph can be translated as seen in Fig. 4.*

5 Related Work and Conclusion

Model transformation is an important concept in order to establish a consistent relationship between source and target models, like business and IT models in enterprise modelling [2]. Security constraints can be defined separately for source and/or target models, but up to now, it is an open problem how to establish a consistent relationship between source and target constraints.

Triple graph grammars have been successfully applied in several case studies for bidirectional model transformations, integrations and synchronizations [15, 12, 16, 10], and there are a variety of formal results concerning correctness, completeness and termination [5], functional behaviour and optimization with respect to the efficiency of their execution [11].

Previous studies on the relationship between model transformations and constraints focussed on general properties, e.g. in order to provide techniques for the verification and validation of model transformations [9, 1, 3] in order to detect underspecified parts or mismatches to the requirements of the domain. They perform semi-automated reasoning using e.g. the theorem prover Isabelle/HOL [9], Prolog [1] or they use OCL validation tools [3]. In contrast to them, this paper has its main focus on a constructive approach for the translation of domain and model specific source into target constraints, i.e. those properties of a source model which are usually not valid for all models of the source language.

Given a model transformation *MT* based on triple graph grammars, we have defined in this paper *MT*-consistent constraints and shown how to propagate source constraints to integrated and target constraints. In our first main result, we prove that under suitable conditions this propagation is consistent in the sense that validity of the source constraint for the source model implies validity of the integrated and target constraint for the integrated resp. target model. Since constraints are often incomplete models we study in Sec. 4 also the propagation of partial *MT*-consistent source constraints leading to our second main result. For this purpose we provide a new concept of model transformations with borrowed context, which allows to transform also model fragments. Transformations of model fragments along total matches are considered for the case of plain graph transformations already in [7] for the refactoring of rules. Based on the new theory of model transformations with borrowed context we will also propagate other kinds of model fragments in future work, e.g. rules of the operational semantics from source to target models in order to prove semantical correctness of model transformations.

Bibliography

[1] Asztalos, M., Lengyel, L., Levendovszky, T.: Towards automated, formal verification of model transformations. In: Proc. ICST '10. pp. 15–24. IEEE (2010)

[2] Brandt, C., Hermann, F.: How Far Can Enterprise Modeling for Banking Be Supported by Graph Transformation? In: Proc. ICGT'10. LNCS, vol. 6372, pp. 3–26. Springer (2010)

[3] Cabot, J., Clarisó, R., Guerra, E., de Lara, J.: Verification and validation of declarative model-to-model transformations through invariants. J. Syst. Softw. 83, 283–302 (2010)

[4] Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.: Fundamentals of Algebraic Graph Transformation. EATCS Monographs in Theor. Comp. Science, Springer (2006)

[5] Ehrig, H., Ermel, C., Hermann, F., Prange, U.: On-the-Fly Construction, Correctness and Completeness of Model Transformationsbased on Triple Graph Grammars. In: Proc. MODELS'09. LNCS, vol. 5795, pp. 241–255. Springer (2009)

[6] Ehrig, H., Hermann, F., Schölzel, H., Brandt, C.: Propagation of Constraints along Model Transformations Based on Triple Graph Grammars. Tech. rep., TU Berlin (2011), to appear

[7] Ehrig, H., Ermel, C., Ehrig, K.: Refactoring of Model Transformations. ECEASST 18 (2009)

[8] Ehrig, H., König, B.: Deriving Bisimulation Congruences in the DPO Approach to Graph Rewriting with Borrowed Contexts. Mathematical Structures in Computer Science 16(6), 1133–1163 (2006)

[9] Giese, H., Glesner, S., Leitner, J., Schäfer, W., Wagner, R.: Towards verified model transformations. In: Proc. Workshop on Model Development, Validation and Verification (MoDeVa 2006). pp. 78–93 (2006)

[10] Giese, H., Wagner, R.: From model transformation to incremental bidirectional model synchronization. Software and Systems Modeling 8(1), 21–43 (2009)

[11] Hermann, F., Ehrig, H., Golas, U., Orejas, F.: Efficient Analysis and Execution of Correct and Complete Model Transformations Based on Triple Graph Grammars. In: Proc. MDI'10 (2010)

[12] Kindler, E., Wagner, R.: Triple graph grammars: Concepts, extensions, implementations, and application scenarios. Tech. Rep. TR-ri-07-284, Department of Computer Science, University of Paderborn, Germany (2007)

[13] Schölzel, H.: Model Transformation of Model Fragments Using Borrowed Context: Extended Version. Tech. rep., TU Berlin, Fak. IV (2010), to appear, online available at http://tfs.cs.tu-berlin.de/publikationen/Papers10/Sch10.pdf

[14] Schürr, A.: Specification of Graph Translators with Triple Graph Grammars. In: Proc. WG'94. LNCS, vol. 903, pp. 151–163. Springer Verlag, Heidelberg (1994)

[15] Schürr, A., Klar, F.: 15 years of triple graph grammars. In: Proc. ICGT'08. LNCS, vol. 5214, pp. 411–425 (2008)

[16] Varró, D., Balogh, A.: The model transformation language of the VIATRA2 framework. Science of Computer Programming 68(3), 214–234 (2007)