

Article

SYSML4TA: A SysML Profile for Consistent Tolerance Analysis in a Manufacturing System Case Application

Sergio Benavent-Nácher , Pedro Rosado Castellano, Fernando Romero Subirón and José Vicente Abellán-Nebot * 

Department of Industrial Systems Engineering and Design, Universitat Jaume I, Av. Vicent Sos Baynat, s/n 7, 12071 Castellón de la Plana, Spain

* Correspondence: abellan@uji.es

Abstract: Tolerance analysis is a key engineering task that is usually supported by domain-specific analysis models and tools that are generally not connected to the system functionality. The model-based system engineering (MBSE) approach is a potential solution to this limitation, but it has not yet been deeply explored in this type of mechanical analysis, for which some problems need to be explored. One of these issues is the capacity of languages such as SysML to describe solution principles based on active surfaces that participate in functionality and are present for tolerance analysis. Thus, this study explored the possibilities that enable SysML to represent these geometries and their mathematical relationships based on Topologically and Technologically Related Surfaces (TTRS) theory and aligned with Geometric Dimensioning and Tolerancing (GD&T) standards. Additionally, the capacity of SysML to assure the consistency of tolerance analysis models is also explored, due to the limitations identified in analysis languages like Modelica. In this context, this paper presents a SysML profile for tolerance analysis modeling (SysML4TA), containing domain-specific semantics (concepts and constraints) to assure the completeness of the analysis models and consistency between the different models considered in the integrated model of the system. Finally, a case study applied to a manufacturing context is presented to validate the capacity of SysML to solve the identified problems.

Keywords: consistency; SysML; OCL; assembly modeling; tolerance analysis; TTRS



Citation: Benavent-Nácher, S.; Rosado Castellano, P.; Romero Subirón, F.; Abellán-Nebot, J.V. SYSML4TA: A SysML Profile for Consistent Tolerance Analysis in a Manufacturing System Case Application. *Appl. Sci.* **2023**, *13*, 3794. <https://doi.org/10.3390/app13063794>

Academic Editor: Nga Nguyen

Received: 20 January 2023

Revised: 10 March 2023

Accepted: 13 March 2023

Published: 16 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The increasing complexity of current technical systems has promoted the adoption of the Model-Based Systems Engineering (MBSE) approach, focused on developing, managing, and controlling models that capture, through different views (structural, functional, behavioral, etc.), the concerns of the involved stakeholders. This approach has been widely applied in multiple fields of engineering, including manufacturing and production systems [1], which are the main context of our proposal. In general, MBSE methodologies handle a heterogeneous set of models due to their management of concepts from different disciplines and domains that belong to different granularity levels and are created with different computer tools and languages. From this heterogeneity, different challenges emerge, for example, those concerning the consistency and completeness of models, as well as the automated transformations between different models.

Many authors argue that consistency assurance is one of today's major industrial challenges [2], both for individual models and for model integration. Although this issue has been widely studied in the software domain, in recent years, it has become more relevant in other domains nearer to production systems, where different issues have gained relevance, such as cross-domain model consistency checking [3], consistency management in automated manufacturing production systems [4], dependency modeling among mechatronic design models [5], etc. Many authors agree that the selection of the language for system modeling is one of the key decisions, according to model categorization. In this sense, two fundamental categories of formal models can be identified: descriptive and analysis

models. On the one hand, descriptive models support design and integration of systems, easing communication among involved stakeholders. Clear examples of these models are UML and SysML models. On the other hand, analysis models are used to mathematically quantify indicators about expected behaviors by running the simulations. Modelica models are typical examples of analysis models, specifically executable simulation models (ESM). However, relationships between these models must be established to eliminate the gap between functional architecture modeling (mainly managed in descriptive models) and physical behavior modeling (managed in analysis models), which is especially important in the use of MBSE for mechanical development, as commented in [6].

In order to narrow this gap, [6] proposes a definition for workflows to connect simulation and design models; a similar solution is proposed in [7], where workflows are used to facilitate collaboration between system architects and analysis experts. Other work lines are explored in [8], where the use of descriptive models to support the design of simulation models is explored to solve some of the deficiencies that most languages and simulation tools present. Specifically, a Modelica simulation model is designed in SysML, taking advantage of the synergy between both languages to build simulation models for multistage manufacturing systems [8], where the logistic flow simulation is enriched with geometric deviation propagation based on the Stream of Variation (SoV) model [9].

An integrated application of the MBSE can also solve some of the deficiencies that most languages and simulation tools show. The combination of heterogeneous models requires the assurance of consistency between the different involved models. However, typical languages for mathematically based analysis models lack mechanisms to support domain-specific semantics, which are necessary to assure model consistency. For example, when focusing on the tolerance analysis models, the consistency should be validated while considering concepts related to artifact geometry and its mathematical definition, both for individual geometries and their relationships. Although the syntax and semantics of simulation languages such as Modelica are perfectly defined, there are not mechanisms for their extension or specialization in order to support domain-specific semantics.

Alternatively, other general-purpose descriptive modeling languages, such as the Systems Modeling Language (SysML) [10], support mechanisms for the development and use of domain-specific modeling languages (DSML), which detail the semantics for a specific domain. Specifically, SysML enables the definition of profiles that extend the language, adding stereotypes and details to the specific semantics through Object Constraint Language (OCL) expressions [11]. These OCL rules, defined in the profile level, can be validated in those constructions of the user model where stereotypes are applied by checking if the rule condition is satisfied through queries without side effects.

SysML has usually been linked to models with a high abstraction, far from the concreteness adopted in typical mechanical engineering models (such as CAD). This is the main reason for the limited adoption of SysML in the field of mechanical engineering [12]. Some works in recent years have explored the integration of SysML models with other domain-specific models, as in [13] or [14], or the mapping of domain-specific models into a common generic system model supported by SysML [15]. Other works are focused on the definition of SysML profiles and, specifically in the domain of mechanical engineering, several proposals such as SysML4Mechatronics [16], SysML4FMArch [17], GERTRUDe [18], or “C&C-M for SysML” [19,20] have been published. However, these works deal with embodiment from the conceptual or functional perspective, but we found no works that used SysML (or SysML profiles) capabilities to model tolerances and support simulation models for geometric variability analysis, an issue that is explored in this work.

The previous review and comments raise the following research questions:

RQ1. Does SysML have enough capability to support a tolerance analysis using the TTRS-based modeling of nominal and deviated geometries?

RQ2. What OCL rules are necessary for assuring the consistency between models according to tolerance analysis semantics? Are OCL rules sufficient to assure consistency?

RQ3. How can profile architecture and defined stereotypes ease subsequent model transformations from SysML to other analysis languages, such as Modelica?

RQ4. What are the necessary steps (procedure) to support analysis modeling? How is this procedure integrated with other previous tasks, corresponding to the system functional and logical architecture modeling?

To explore these issues, our research group is developing the framework illustrated in Figure 1; it is based on a SysML central model (optionally constructed from other CAx models) and the use of model transformations. In this framework, two goals guide the use of SysML profiles: (a) consistency assurance between models according to specific domain semantics (e.g., SysML4TA) and (b) concept identification to facilitate model-to-model and model-to-text transformations (e.g., “SysML4TA for Modelica”). This paper is focused on the first goal, specifically, on the definition and development of the SysML for Tolerance Analysis (SysML4TA) profile.

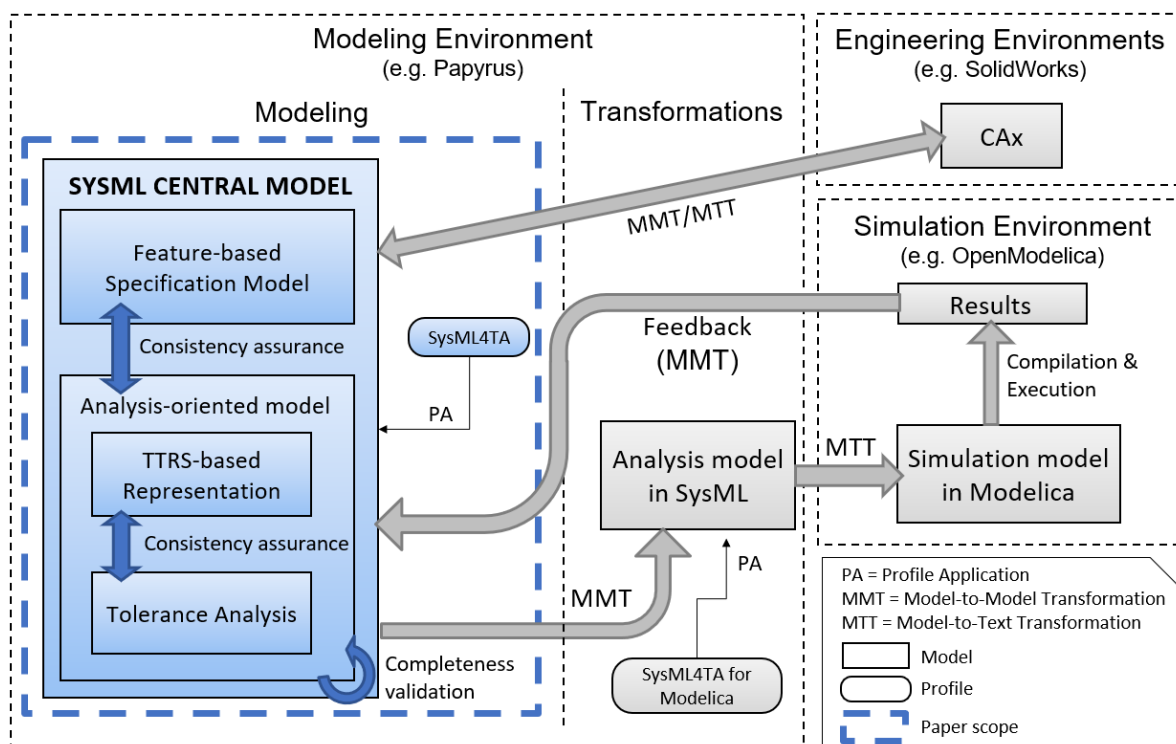


Figure 1. Proposal framework.

The theoretical framework of this proposal was established from a literature review of previous proposals focused on two types of models. On the one hand, models that support the specification/design of assemblies were reviewed, including the metamodels proposed by the NIST (core product model (CPM) [21] and the Open Assembly Model (OAM) [22]) or additional proposals based on them (such as Modified OAM [23]). These metamodels provide a standard representation of assemblies, facilitating interoperability in CAx environments. On the other hand, we also considered models that facilitate the principle solution by, e.g., supporting the modeling of active surfaces and their relationships. In this group, the Contact and Channel Approach (C&C²-A) [24], as well as other works around it (such as [19] or [20]) was reviewed. These models can support decisions about the appropriateness of principle solutions based on qualitative analysis, but they cannot support quantitative analysis. For this reason, we explored the Technologically and Topologically Related Surfaces (TTRS) model [25] to represent the geometry of mechanical artifacts. Moreover, to support nominal definitions and the computation of deviated geometries, we combined TTRS concepts with a small displacement torsor (SDT) [26] model, as described in detail in the next sections.

The content and structure of this document is summarized below. Section 2 contains the fundamentals of the TTRS model and the main mathematical considerations for supporting the modeling of tolerances. Section 3 is dedicated to the methodological proposal, including a description of the SysML4TA profile, a proposed library (TA_Library), and a suggested procedure to define user models for tolerance analysis. Section 4 presents a case study to exemplify the application of the proposal and validate its suitability in the context of manufacturing by analyzing a functional condition in manufacturing equipment that influences the geometric quality of the product. Finally, Section 5 contains a final discussion and Section 6 summarizes our main conclusions and recommendations for future works.

2. TTRS and Mathematical Modeling

2.1. Technologically and Topologically Related Surfaces (TTRS)

As mentioned above, we adopted the TTRS model [25] as the main reference to support tolerance analysis models. This paper proposes an interpretation of the theory and a definition of the TTRS concept that is compatible with the definitions provided by other authors, but that presents a clear orientation for SysML modeling.

The TTRS model is based on the geometric invariance class concept, which classifies surfaces into seven types, among which, the three translation and three rotation degrees of freedom (DoF) can be variant (in which a change modifies the position of the surface in the space) or invariant (in which a change keeps the surface invariant). The seven invariance classes are spherical, cylindrical, planar, helical, revolute, prismatic, and complex. Each invariance class has a minimum set of reference geometric elements, named Minimum Geometric Reference Elements (MGREs). Each MGRE is composed of one, two, or three Reduced Geometric Elements (RGEs), which can be a point, line, or plane, that fulfill some geometrical constraints between them, depending on the invariance class.

From a technological and topological perspective, surfaces are usually associated. A TTRS is an assembly (set) formed by two surfaces (or a surface and a TTRS, or two TTRSs) belonging to the same solid (topological aspect) and being located in the same kinematic loop in a given mechanism (technological aspect). Moreover, many authors have also transferred this definition to the relationships between surfaces of different solids, introducing the concept of Pseudo TTRS [18].

A TTRS associates two surfaces through the definition of constraints between their MGREs (or between the RGEs owned by each MGRE). These associations are characterized, giving rise to 44 cases of combinations of surfaces according to the invariance class of each of them and the possible constraints imposed between them. Each of these combinations gives rise to a whole characterization through a new invariance class (reclassification mechanism).

These 44 cases are shown in Figure 2, where a color code is used to differentiate three particular sets of MGRE associations: association between simple MGRE, composed by one RGE (green); associations with at least one complex MGRE, composed by more than one RGE (yellow); and associations where a helical MGRE participates (gray), which are identified separately, because they are not considered in this work (hence, only the remaining 35 cases were modeled). Differentiation between green and yellow associations is important, because cases with complex MGREs are supported by a combination of different constraints defined in the 13 simple cases.

From the perspective of our proposal, the analysis of pairwise surfaces and the constraints between them is very interesting for the representation of geometric specifications and tolerances, as has been explored in [27].

TTRS is suitable for representing tolerance zones (TZ) according to geometric dimensioning and tolerancing (GD&T) standards [28], especially basic cases such as planar, cylindrical, or spherical TZs. These TZs can be defined by their corresponding MGRE (according to their invariance class), and their size (basically width or diameter). According to GD&T standards, more complex TZs can be constructed as a result of the intersection or union of simple TZs.

		MGRE						
		Spherical	Planar	Cylindrical	Helical	Revolution	Prismatic	Complete
MGRE	 Spherical	1-coincidence 2-otherwise	3-always	4-coincidence 5-otherwise	14	20-coincidence 21-otherwise	29-always	38-always
	 Planar		6-parallel 7-otherwise	8-perpendicular 9-parallel 10-otherwise	15	22-perpendicular 23-otherwise	30-parallel 31-otherwise	39-always
	 Cylindrical			11-coincidence 12-parallel 13-otherwise	16 17	24-coincidence 25-otherwise	32-parallel 33-otherwise	40-always
	 Helical				18 19	26	34	41
	 Revolution					27-coincidence 28-otherwise	35-always	42-always
	 Prismatic						36-parallel 37-otherwise	43-always
	 Complete							44-always

Figure 2. The 44 association types between MGREs.

2.2. Mathematical Modeling

As mentioned previously, TTRS is aligned with geometric dimensioning and tolerancing (GD&T) standards. However, it does not support a description of deviated geometries that is detailed enough to quantify their effects on part location/orientation in the assembly. For this reason, TTRS concepts must be enriched with additional mathematical constructions, which are a basic part of analysis models. This work proposes the combination of TTRS concepts with the application of a small displacement torsor (SDT) [26] model to support the computation of deviated geometries. The values of the SDT define the maximum deviations applicable to each variant DoF, calculated according to the bounded geometries and the TZ, depending on its type and size.

The TTRS model references each surface using RGEs, which can be a point, a line, or a plane. These basic geometric elements must be defined by an appropriate mathematical expression: in this case, a combination of points and vectors expressed in a local Cartesian system, specific to each part considered. This local geometric description enables the independent specification of each part, but it is not compatible with the definition of assembly relationships. To this end, all the geometries involved in the analyzed assembly must be expressed with respect to a common global system, enabling the definition of mathematical expressions that support Pseudo TTRSs. For this reason, the local definition of the geometries must be transformed to a global definition, changing the reference system. This transformation can be supported by a transformation matrix that is characteristic of each part. Once all the geometries of every part are expressed with respect to a common system, all TTRS restrictions can be implemented as mathematical expressions between them (fundamentally vector and scalar products), supporting both part specifications and assembly relationships.

Our proposal conforms to the need to model nominal and deviated geometries, as well as expressing them with respect to local and/or global coordinate systems, by adopting the following mathematical processing from the nominal geometry of each part with respect to local coordinate systems: (a) computation of deviated geometries by applying an SDT vector that is compatible with the limits imposed by the TZ and for which the deviated geometries are also expressed with respect to the local coordinate systems; (b) coordinate transformation of deviated geometries to a global coordinate system by applying a paramet-

ric transformation matrix whose values must be calculated from the assembly relationships that restrict the position/orientation of each part.

3. DSML-Based Methodology Proposal

Generally, a methodology is defined as a set of methods (languages and procedures) and tools applied to solve a problem, ideally derived from a theoretical framework and from a particular point of view. In our case, the addressed problem is the consistent model-based analysis of the geometric variability in a mechanical assembly for validating functional conditions, specified in GD&T terms. We address the problem with an MBSE approach in the context of a central model that supports all the information/data and assures consistency between the involved concepts from different viewpoints (assembly representation, TTRS representation, and functional condition analysis).

To support this approach, the SYSML4TA language and a procedure were developed (presented in Sections 3.1 and 3.3, respectively). Additionally, we also developed the TA_library (Section 3.2). The SYSML4TA and TA_library are grouped in a package of modeling resources named the TA_Toolkit, whose content is presented in Figure 3, where the user-level model structure is also shown.

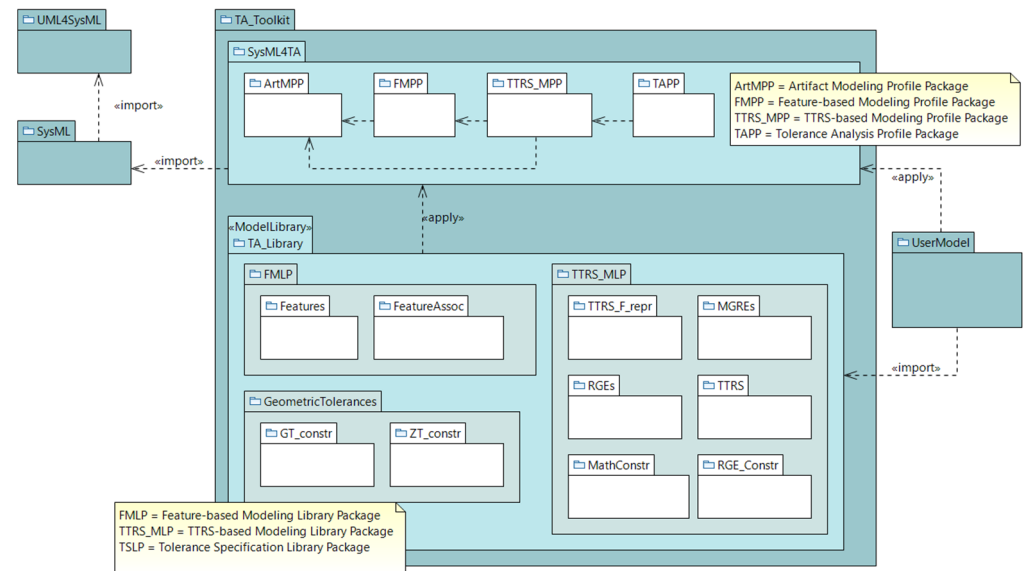


Figure 3. Package diagram of the developed models.

To implement the proposal, Papyrus [29] was selected as the modeling tool, both for the SYSML4TA development and for user-level modeling. Papyrus is an open-source Model-Based Engineering tool for model edition and transformation and is integrated in the Eclipse Modeling Framework. It supports DSML-based approaches by defining and applying UML profiles in a very rich and efficient manner. Moreover, it offers a specific validation view that allows for evaluating OCL expressions to facilitate consistency management. Although the methodology is a tool-independent proposal, and another software that supports SysML modeling and profiling could have been used, Papyrus was selected because it is a widespread choice in academia for teaching and research purposes.

3.1. SysML Profile for Tolerance Analysis (SYSML4TA)

SYSML4TA extends SysML to incorporate the necessary concepts for tolerance analysis, mostly from the assembly domain and TTRS-based modeling. Each of these concepts is defined as a stereotype in the SYSML4TA by extending UML metaclasses and/or specializing SysML stereotypes, as shown in Figure 4. In the figure, stereotypes specializing or extending the same metaclass or SysML stereotype are represented in the same color, and the diagrams that follow it also conform to this color code. The profile is structured with

four packages that correspond to different stereotypes, although the stereotypes of different packages are interrelated (dependencies depicted in Figure 3). One subsection is dedicated to presenting each of these packages, including a brief textual description of each stereotype and the depiction of some conceptual diagrams to represent the meta-model abstract syntax. The concepts represented in these diagrams were implemented as stereotypes in the profile (with the same names), and most of the relationships depicted (compositions, dependencies, etc.) are restrictions of the meta-model that were implemented as OCL rules. A complete list of definitions of the stereotypes can be found in Appendix A, and the implementation of some of the OCL expressions is exemplified in Appendix B.

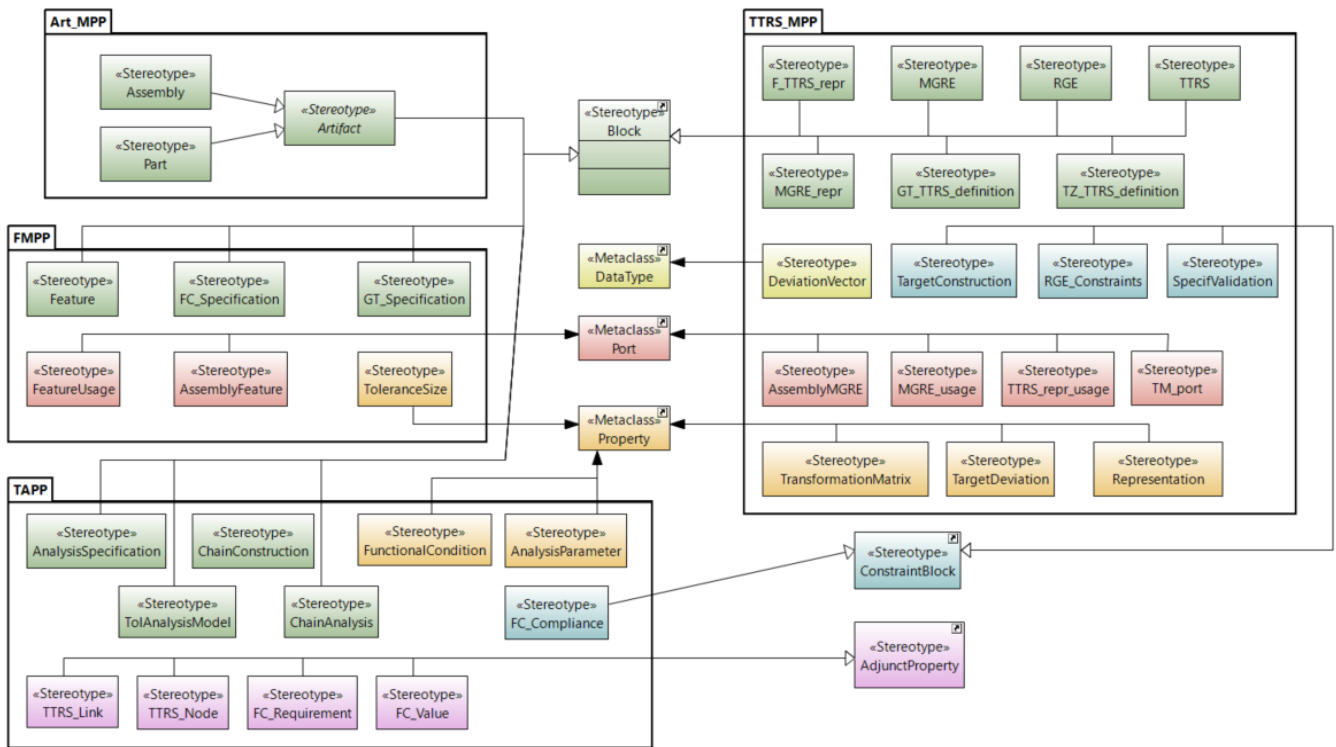


Figure 4. Package diagram with the stereotypes of the developed profile.

3.1.1. Artifact Modeling Profile Package (ArtMPP)

ArtMPP includes stereotypes to support the assembly modeling of the system, describing the assembly structure based on the abstract concept of Artifact and its specializations, Part and Assembly. As represented in Figure 5, an Assembly is composed of at least two artifacts (Parts or Subassemblies). Moreover, any Artifact contains Features (specification of a shape portion of an Artifact) and their F_TTRS_repr, that is, the TTRS-based representation of each Feature in an Artifact. Although these concepts are developed in the next sections, it is important to clarify that the Features are used for the geometric specification of the artifact, while F_TTRS_repr are used for analysis-oriented modeling. To support the strong relationship between properties related with these two views, the stereotype «Representation» is defined as an extension of the Property metaclass. This stereotype includes the “representsTo” Attribute pointing to another property owned by the same parent. Using this mechanism, a property typed by a F_TTRS_repr must be stereotyped by «Representation», defining its “representsTo” Attribute as a property typed by a Feature owned by the same Artifact.

3.1.2. Feature-Based Modeling Profile Package (FMPP)

FMPP includes the stereotypes to support the artifact geometric specification based on the Feature concept (similar to that proposed in CPM/OAM) and relationships (associations) between Features. Contrary to the OAM model, this proposal adopts a unified

conceptualization of feature associations both for intra-part and inter-part feature associations. As depicted in Figure 6, an Artifact is composed of Features, resulting in parts typed by a «Feature» Block to define part geometric specifications. Additionally, some of these parts are connected to delegated Ports («AssemblyFeatures» Ports), which are also typed by «Feature» Block. These Ports support assembly relationships between Artifacts according to the encapsulation principle.

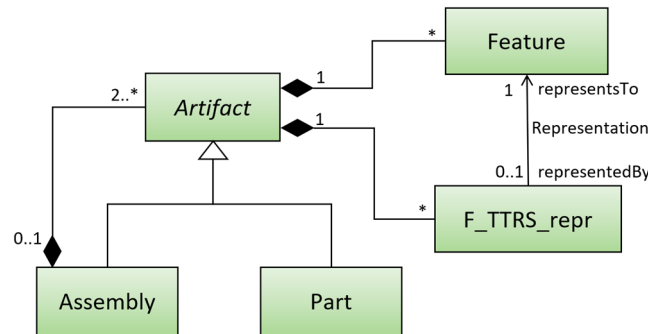


Figure 5. Abstract syntax of artifact definition (“*” is a “zero to many” multiplicity).

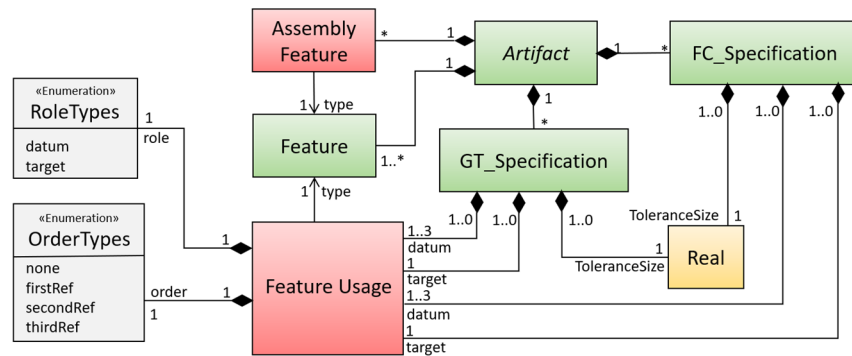


Figure 6. Abstract syntax of feature-based artifact specification (“*” is a “zero to many” multiplicity).

To support the wide range of GD&T standard-compliant specifications, geometric tolerance (GT) specifications were defined both for binary associations between two features and more complex specifications for which more than one datum was required. In a similar way, the FC_Specification concept represents a relation between Features that defines a functional condition to be analyzed. Although both concepts have an associative nature (relating datum and target features), these stereotypes are modeled as Block specializations that are characterized by «FeatureUsage» Ports (1 target feature and 1 to 3 datum features), and a «ToleranceSize» Property (Real that represents the size of the TZ). For a complete definition of «FeatureUsage» Ports, this stereotype includes two Attributes specifying its role (datum/target) and the order of the datum (“none” if it is a target). Association Blocks (AssociationClass stereotyped as Block) were avoided due to their limitations in supporting non-binary relationships, as well as relationships in which it is essential to differentiate the roles of each associated entity (which Feature is datum and which is target, for example).

Finally, it is important to clarify that the Feature-based view of each Artifact is used as a descriptive modeling of the geometric specification, adopting a system view. The analysis-oriented modeling and the mathematical definition of the geometries and GTs are supported by the stereotypes of the following package.

3.1.3. TTRS-Based Modeling Profile Package (TTRS_MPP)

TTRS_MPP includes stereotypes to support the analysis-oriented representation of the system based on TTRS concepts, that is, the mathematical definition of the position and orientation of the geometries according to their invariance classes. As proposed in the TTRS model, each invariance class is characterized by a particular MGRE, and a MGRE is

composed of one, two, or three RGEs and RGE_constraints (ConstraintBlock specializations) between them when necessary (Figure 7).

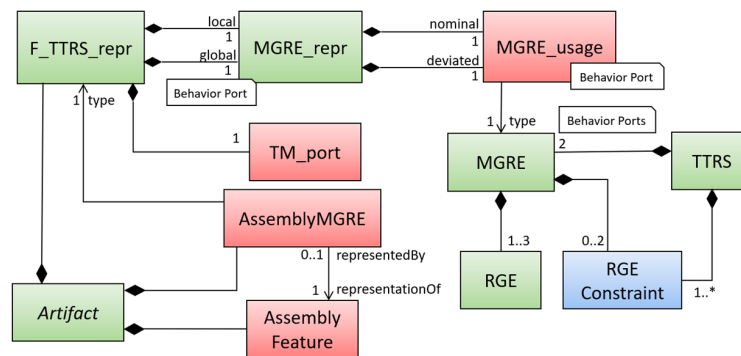


Figure 7. Abstract syntax of TTRS-based concepts (“*” is a “zero to many” multiplicity).

The TTRS-based representation of a feature (F_TTRS_repr) includes the “local” property, typed by MGRE_repr, that contains the mathematical definition of the feature with respect to the local coordinate system. This MGRE_repr owns two «MGRE_usage» Ports typed by MGRE, one for the nominal definition and other for the computed deviated definition. As mentioned in previous sections, MGREs must be also expressed with respect to a global coordinate system that is common for all the parts participating in the assembly. For this reason, the F_TTRS_repr also includes the “global” Port, typed by MGRE_repr, which is computed from “local” through a coordinate transformation. In this proposal, the transformation is governed by an homogeneous transformation matrix owned by the part. This matrix is defined as a «TransformationMatrix» property of each part, which is connected to the «TM_Port» Port of an «F_TTRS_repr» Block.

Moreover, to characterize the 35 combinations of related MGREs considered in this work, the TTRS concept was also included as an entity with two Ports typed by MGREs and the mathematical constraints between their RGEs (RGE_constraints). Another important concept is the AssemblyMGRE Port, which is the TTRS-based representation of the previously described AssemblyFeature Port. «AssemblyMGRE» Ports are always typed by «F_TTRS_repr» Blocks, and they support the assembly relationships through connection between TTRS-based representations of different Artifacts. The set of concepts and relationships mentioned up to this point are represented in Figure 7.

TTRS_MPP also includes the mathematical definition of geometric tolerances (GT_TTRS_definition), supporting the equations that define a tolerance zone (TZ_TTRS_definition), limiting the position and orientation deviations in target geometries (MGREs) with respect to datum geometries. It is important to point out that form deviations/tolerances were not considered in this work.

As depicted in Figure 8, GT_TTRS_definition is composed of: (a) TZ_TTRS_definition; (b) a Port of type TTRS_repr_usage named “target”; (c) 1 to 3 Ports of type TTRS_repr_usage named “datum” (d) mathematical constraints (TTRS) to construct the TZ_TTRS_definition with respect to the “datum” Ports; (e) a ConstraintBlock to validate the correctness of the GT specification (SpecifValidation); (f) a ConstraintBlock to constructs a deviated target (TargetConstruction) with respect to the TZ_TTRS_definition; (g) a deviation vector (TargetDeviation), whose components specify the deviation value in each variant DoF.

Regarding the commented ConstraintBlocks, it is important to clarify that these constructions include as constraints the equations defining the mathematical relationships between geometries, that is, MGREs and its RGEs. Specifically, SpecifValidation defines equations by combining some of the 13 simple cases of MGRE associations, while TargetConstruction defines equations to construct deviated geometries according to the values of the considered SDT.

As in the FMPP previously discussed, although some of the concepts have an associative nature, they have been modeled as Block specializations. The set of concepts and relations mentioned before are depicted in Figure 8.

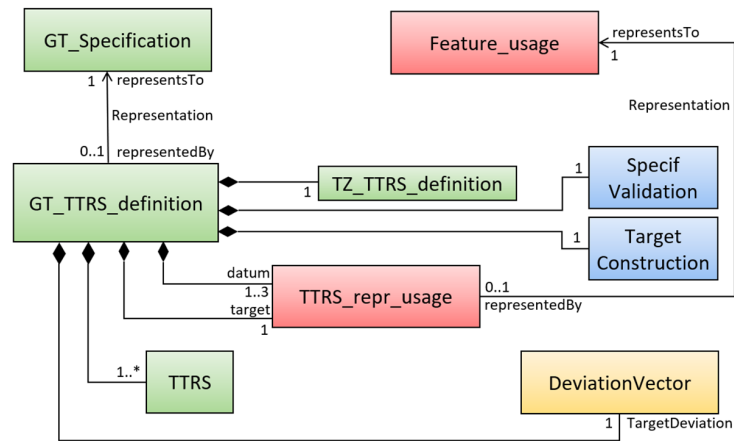


Figure 8. Abstract syntax of TTRS-based geometric tolerance representation (“*” is a “zero to many” multiplicity).

3.1.4. Tolerance Analysis Profile Package (TAPP)

TAPP includes the stereotypes to create the tolerance analysis model of an assembly, which can include one or more chain analyses. Each chain is constructed to quantify the variability of a geometrical characteristic derived from a functional or assembly requirement, named the Functional Condition; accordingly, the chain to study a Functional Condition is named the Functional Chain. Each Functional Chain is constructed on a specific assembly configuration, which sometimes requires additional constraints to block all assembly DoFs. A Functional Chain is composed of a sequence of links corresponding to the GT that relates different geometries (MGREs), acting as nodes, and a Functional Condition link that closes the chain. Chain elements, links, and nodes are all defined as properties, stereotyped as AdjunctProperty specializations, whose Principal Attribute points to an element existing in the TTRS-based representation model.

The Chain Analysis also includes at least one Analysis Specification, where the input Parameters (DeviationVector usages and FC_Requirement) for the analysis are established. The analysis requires as many DeviationVectors as nodes included in the chain, relating nodes’ properties and vectors through a ConstraintBlock that establish this assignment. Resulting from the Chain Analysis, the value for the Functional Condition is obtained (FC_Value). Finally, the FC_Compliance ConstraintBlock establishes the equations for the comparison between FC_Value and FC_Requirement.

Stereotypes that support the TA are presented in Figure 9 and exemplified in the case study presented in Section 4.

3.2. Tolerance Analysis Library (TA_Library)

To facilitate model creation at the user level, a library was developed. The description of this library and its well-defined reusable constructions exemplify the application of some of the previously described stereotypes, facilitating semantic understanding. The library contents were structured in three main packages, but the library can be extended by developers to include other reusable constructions.

3.2.1. Feature-Based Modeling Library Package (FMLP)

FMLP includes different types of features and feature associations for assembly modeling. As shown in Figure 3, this package is structured in three sub-packages, whose contents are presented in Table 1.

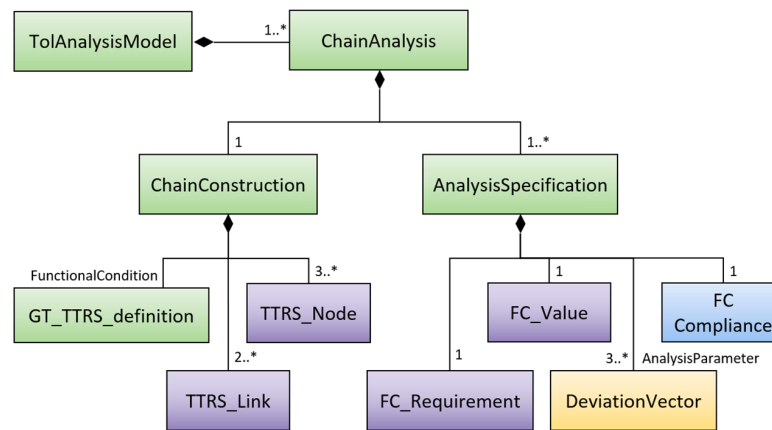


Figure 9. Abstract syntax of tolerance analysis model (“*” is a “zero to many” multiplicity).

Table 1. FMLP sub-packages and contents.

Package	Description	Some Constructions
Features	Blocks stereotyped as «Feature». Each Feature can also contain Attributes (Value Properties) related to geometry (diameter of an axis, for example) and necessary information for the analysis.	PlanarFeature CylindricalFeature ...
Feature Associations	Blocks stereotyped as «FeaturesAssociation» to support different types of associations between Features, both intra-part specifications (relative position, relative orientation, etc.) and inter-parts relationships to support assembly contacts and fittings (planar contact, hole-axis fitting, etc.).	Distance_parallel_planes PlanesPerpendicularity PlanarContact Hole-Axis_fitting ...

3.2.2. TTRS-Based Modeling Library Package (TTRS_MLP)

TTRS_MLP includes reusable constructions for the TTRS representation. Excluding the helical MGRE, six MGREs are defined as a set of up to three RGEs (point, line, and/or plane) related by specific constraints, according to the invariance class. «TTRS» blocks are also defined to support the 35 possible combinations between these six MGREs. The TTRS_MLP is structured by different sub-packages, whose constants are described in Table 2.

As an example of a «TTRS» Block definition, Figure 10 shows two diagrams related to Case6 block that associates two parallel planar MGREs. Figure 10a shows the Case6 PAR diagram to represent the connections between Parameters of the C6 ConstraintBlock and different properties. Finally, Figure 10b shows the PAR diagram of C6 ConstraintBlock, which is composed of two ConstraintBlocks to mathematically ensure the parallelism and distance between the two planes. Each ConstraintBlock contains a constraint that defines mathematical non-causal equations using the Modelica language.

PlanarF_Repr is described as an exemplification of a TTRS-based representation. As shown in the IBD of Figure 11c, a PlanarF_Rep owns two MGRE_planar_repr properties (Local and Global) and one «TM_port» Port of type Matrix. Local and Global properties are related as shown in the PAR diagram (Figure 11d) by means of a coordinate transformation matrix. A MGRE_planar_rep is formed by two properties of type MGRE_Planar (nominal and deviated) as shown at the IBD of Figure 11b. Finally, the IBD at Figure 11a shows that a MGRE_Planar is composed of a behavior Port typed by RGE_Plane Block, which owns two nested ports corresponding to the point and the normal vector.

3.2.3. Tolerance Specification Library Package (TSLP)

The TSLP offers a repository of constructions to model the most typical GTs in GD&T specifications. To structure the library, specific packages were defined to organize the GT constructions according to the type of target element to be controlled. These packages include both the specification of GT between Features (XXX_specif) and the mathematical definition of GT between MGREs (XXX_cons). Since these constructions use common

elements, two additional packages were defined, containing the simple TZs and the specific Blocks for validation and construction of the target element (V&C Blocks). Table 3 describes these packages and some of their contents.

Table 2. TTRS_MLP sub-packages and contents.

Package	Description	Some Constructions
TTRS_F_repr	Repository of Blocks that support TTRS-based representations for the different features, including two MGRE_repr to support the local and deviated definition, and the corresponding transformation.	PlanarF_repr CylindricalF_repr ...
MGREs	Blocks stereotyped as «MGRE». The TTRS theory identifies 7 different MGREs, corresponding to the 7 invariance classes, but the MGRE associated with the helicoidal invariance class has not been considered in this library.	MGRE_spherical MGRE_planar MGRE_cylindrical ...
MGRE_repr	Repository of Blocks that contains nominal and deviated MGREs to represent a surface according to its invariance class.	MGRE_spherical_repr MGRE_planar_repr ...
TTRS	Blocks defined to support the association between MGREs considered in the previous package. These constructions are stereotyped as «TTRS», and based on the TTRS theory, 35 types of TTRS were defined, excluding cases with helicoidal MGREs (see Figure 2). Each TTRS contains at least one RGE Constraint.	Case1 Case2 ... Case44
RGEs	Blocks stereotyped as «RGE» to support the reduced geometric elements. Based on the TTRS theory, three types of RGEs have been defined (point, plane, line).	PointRGE PlaneRGE LineRGE
RGE Constraints	ConstraintBlocks, stereotyped as «RGE_Constraint», with almost two RGE Parameters. Based on the TTRS theory, 13 constraints were defined, aggregating some additional ConstraintBlocks from the next package.	C1 C2 ... C13
Basic Mathematical Constraints	Additional ConstraintBlocks to support geometric operations between points, vectors, or matrices, for example.	Perpendicular_vectors Point_Contained_in_line ...

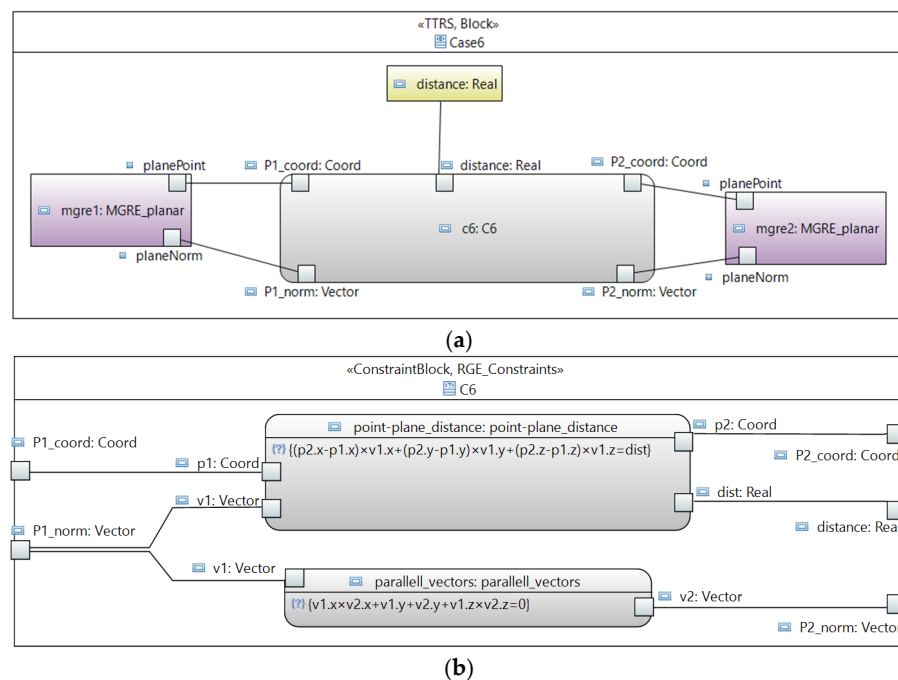


Figure 10. Diagrams related to TTRS Case6: (a) PAR of TTRS block Case6 «TTRS» block (planar parallelism); (b) PAR of the C6.

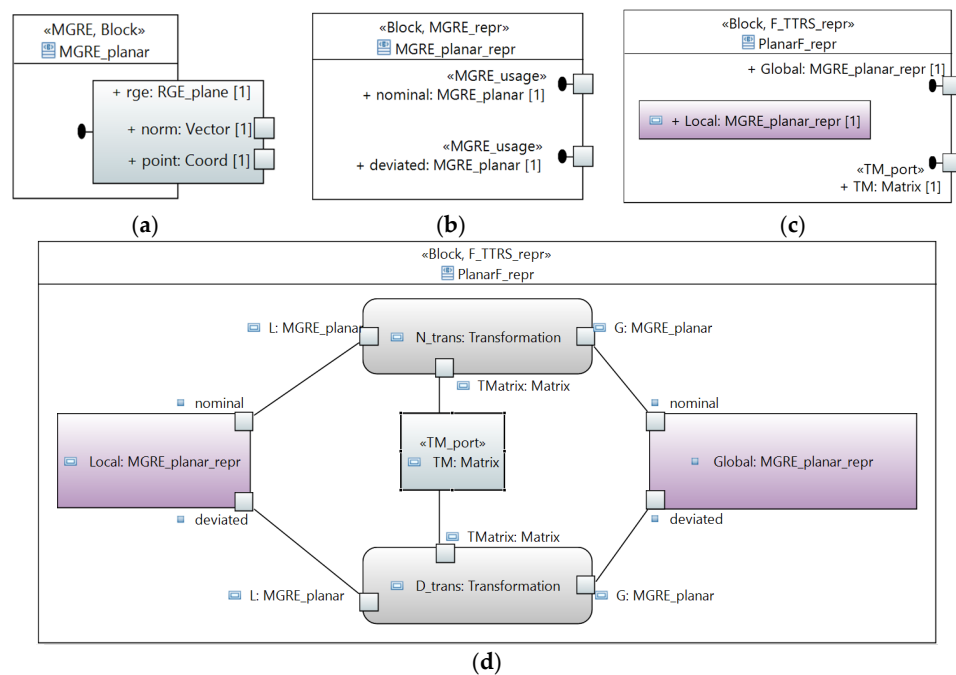


Figure 11. Diagrams related to TTRS-based representation for planar feature: (a) IBD of MGRE_planar block; (b) IBD of MGRE_planar_repr block; (c) IBD of PlanarF_repr block; (d) PAR of PlanarF_repr block.

Table 3. TSLP sub-packages and contents.

Package	Description	Some Constructions
GT4PlanarTarget	Package containing geometric tolerances for which the target element is a planar geometry. The unique possible TZ in this case is the planar tolerance zone, because it is the only zone type capable of controlling the position and/or orientation of a plane.	Plane-Plane_Distance_specif Plane-Plane_Perpendicularity_specif Plane-Plane_Distance_cons Plane-Plane_Perpendicularity_cons ...
GT4AxialTarget	Package containing geometric tolerances for which the target element is a cylindrical geometry. These GT can use different tolerance zones, such as planar, cylindrical, or prismatic TZ, according to the number of references.	Plane-Axis_Distance_specif Axis-Axis_Coaxiality_specif Plane-Axis_Distance_cons Axis-Axis_Coaxiality_cons ...
GT4PunctualTarget	Package containing geometric tolerances for which the target element is a punctual geometry. These GT can use different tolerance zones, such as planar or spherical TZ, according to the type and number of references.	Plane-Point_Distance_specif Axis-Point_Distance_specif Axis-Point_Distance_cons ...
Simple Tolerance Zones	Package containing simple tolerance zones, each one being characterized by a MGRE and a size.	TZ_Planar TZ_cylindrical ...
V&C blocks	Package containing blocks that relate the target element with the tolerance zone, containing the necessary constraints to support the validation of the specification correctness and the constructions of the deviated target.	Plane_in_planarTZ Cylinder_in_planarTZ Cylinder_in_cylindricalTZ ...

Although this library version only includes simple TZs, the library can be extended to include more complex TZs, for example, a prismatic tolerance zone modeled as the union of two planar tolerance zones.

The Distance_parallel_planes_cons is an exemplification of GT construction. This block is composed of different properties, connected as shown in the IBD of Figure 12a. “D” port (typed by PlanarF_repr) is the datum of this GT. “TZ_NominalDef” and “TZ_DeviatedDef” (typed by Case6) contain the constraints to construct the nominal and deviated MGREs of

the planar TZ, stored in the “tz” property. “V&C” property (typed by V&C_planarTZ_plane) contains the constraints for the validation of the specification and the construction of the deviated target (“T” Port), applying the deviations defined in the “devVector” property. Figure 12b presents the PAR diagram of the V&C_planarTZ_plane Block, showing the connections between its properties and the Parameters of its ConstraintBlocks (“validation” and “construction”).

3.3. Procedure

This section presents the procedure followed in the proposed methodology for the creation of a tolerance analysis model for a particular system. This analysis model must be constructed from the specification of the Artifact, that is, the assembly model of the system. This assembly model can be the result from a design process with system modeling tools or can derive from previous detailed design models (CAD, for example). In both cases, the proposed procedure involves three main steps, corresponding to the views/perspectives considered: feature-based assembly model, TTRS-based representation model, and TA model.

3.3.1. Assembly Modeling

This step is focused on specifying the system’s hierarchical structure from an artificial point of view, including feature-based geometry description. This modeling process implies the system description using BDDs and IBDs. BDDs are used to show the different aggregation levels and composition relationships, including features and the GT and FC specification between them. The IBDs are used to show both the internal relations between the features and the GT and FC specifications and the delegated ports that offer some features to the environment for assembly relationships. We recommended diagramming at least one BDD and one IBD for each Artifact.

3.3.2. TTRS-Based Modeling

During this second step, the previous specification model is enriched to interpret the system architecture from a TTRS perspective, and it must be consistent with the previous one. This means that all «Feature» and «GT_Specification» Blocks must have their TTRS-based representation, modeled as «F_TTRS_Repr» and «GT_TTRS_deviation» Blocks, respectively. Again, to represent the TTRS structure, BDDs and IBDs are used. The new TTRS-based composition relationships must be represented in at least one BDD, and a new IBD for each «Artifact» Block must be created from the TTRS perspective.

In the case of the TTRS representation, each previously defined «Part» Block is also composed of «F_TTRS_Repr» Blocks (shown in BDDs). Again, the strategy of using delegated Ports («AssemblyMGRE») is applied. «GT_TTRS_deviation» Blocks are defined between TTRS-based representations (Properties or «AssemblyMGRE» Ports typed by «F_TTRS_Repr» Blocks), and they provide the mathematical interpretation of «GT_Specification» Blocks defined in the previous step. In this sense, TTRS-based Properties are stereotyped as «Representation» to point to the corresponding Feature-based Properties. All these elements and connections should be represented in an IBD of each «Artifact» Block.

PAR diagrams are less necessary during this step if the user uses the library constructions, because the constraints for the mathematical interpretation are already implemented at the library level, where PARs are already available.

3.3.3. Tolerance Analysis Modeling

Finally, the third step focuses on the construction of the analysis model. This analysis model includes a single «TolAnalysisModel» Block that is composed of several «ChainAnalysis» Blocks. For each «FC_Specification» to be analyzed, a «ChainAnalysis» Block is created, including a «ChainConstruction» Block and one or more «AnalysisSpecification» Blocks, where different inputs can be defined depending on the analysis strategy. Each «AnalysisSpecification» Block has a shared relationship to the «ChainConstruction»

Block to analyze the chain with different input values or strategies. This structure must be represented in at least one BDD diagram.

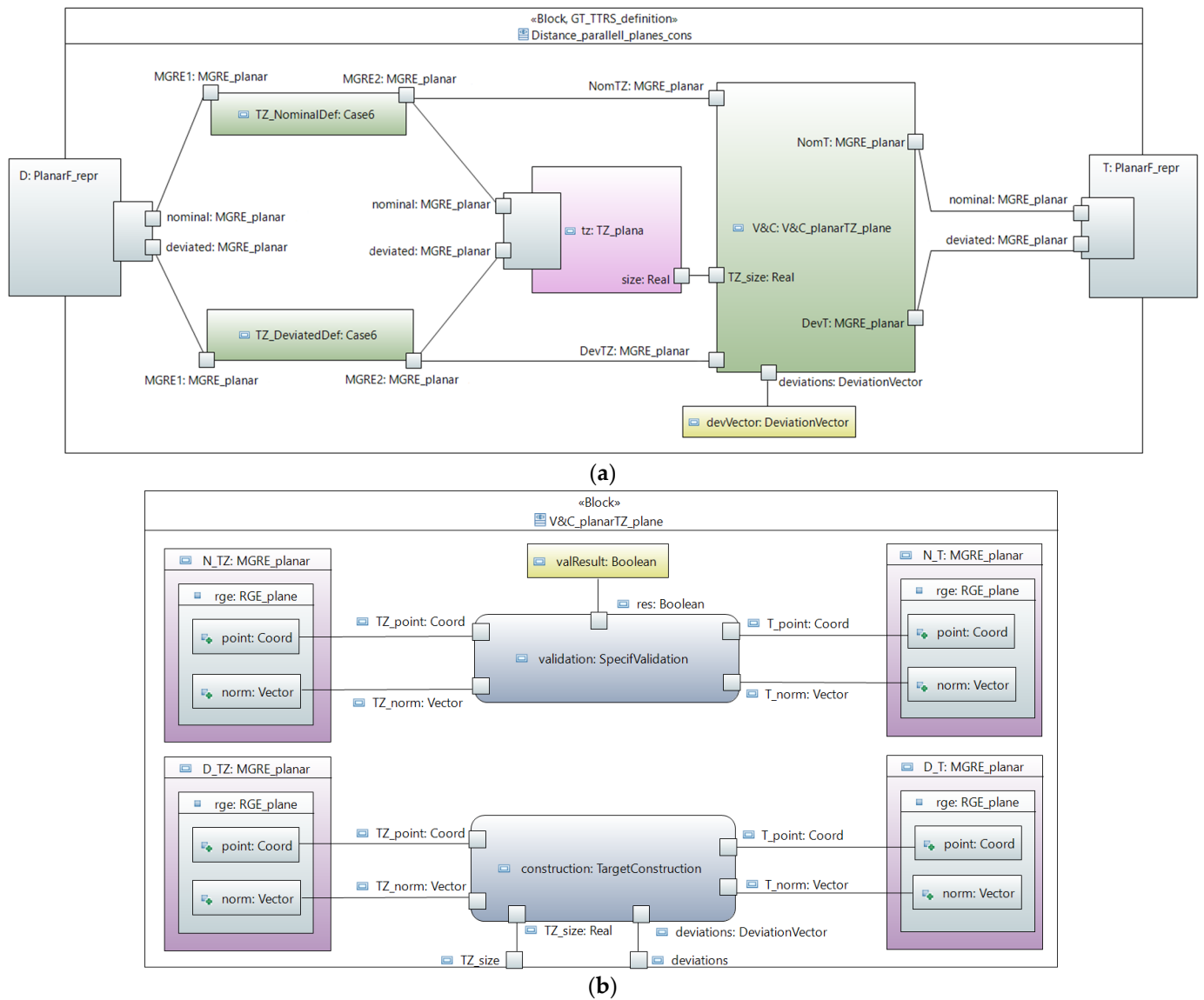


Figure 12. (a) IBD diagram of the Distance_parallel_planes_cons Block; (b) PAR diagram of the V&C_planarTZ_plane block.

On the one hand, the «ChainConstruction» Block includes a set of «TTRS_Node» and «TTRS_Link» Properties, a Property typed by a «Functional Condition» Block, and additional constraints to establish the assembly configuration, if necessary. An IBD of each «ChainConstruction» Block should be defined to graphically represent the chain. It is necessary to remember that «TTRS_Node» and «TTRS_Link» properties are AdjunctProperties whose Principal Attribute points to «F_TTRS_Repr» and «GT_TTRS_deviation», respectively.

On the other hand, the «AnalysisSpecification» Blocks are composed of different «AnalysisParameter» Properties and at least one «ParamAssignment» ConstraintBlock for the assignment of input values to internal Properties of each «TTRS_Link». These assignments must be shown in a PAR diagram for each «AnalysisSpecification» Block.

4. Case Study

The use of the profile and the proposed library is exemplified through a case study to answer the following questions: (a) are the defined stereotypes sufficient to completely

define the concepts handled during the analysis model construction? (b) do the constraints implemented detect inconsistencies between system and analysis model? (c) is analysis model completeness validated through the considered constraints? (d) does the proposed methodology facilitate the modeling work? (e) what are the main advantages and disadvantages of the proposed methodology with respect to an analysis model defined independently (without relations with a system model)?

The case study was inspired by a coinage process presented in [30], so the proposal was applied during the design of a manufacturing system. The manufacturing system was an assembly composed of a press, a punch system, and a die system, as shown in Figure 13. Although the complete Artifact was modeled, the proposed analysis was limited to study the Functional Conditions between the die tool and punch tool and specifically, the deviation in the relative position between the two planar faces that imprint the coin relief (Figure 13). Although these faces are nominally parallel in this assembly configuration, the deviations of these real surfaces of the manufacturing equipment directly influence the geometric quality of the manufactured product (coin), as depicted in Figure 13. Due to this limitation of the case study, this section illustrates the model developed only with some of the most representative or exemplifying diagrams, but the complete model is not provided.

The following three subsections briefly describe the modeling process, corresponding to the three steps of the procedure described in Section 3.3. Due to space limitations, only a few diagrams are shown in each section to graphically show some examples of modeled elements. A color code was defined to identify each type of element represented in the diagrams: Artifacts (blue), Feature/TTRS-based Feature representation (purple), GTs (green), Functional Condition (orange), mechanism configuration constraint (red), and analysis model and Functional Chains (white).

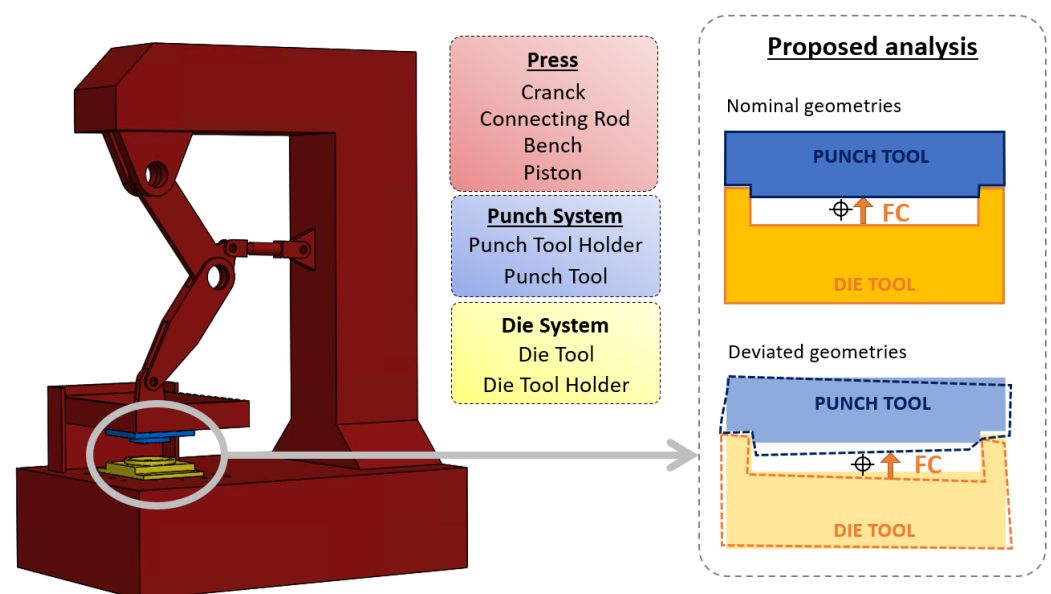


Figure 13. CAD representation of the coinage system and the Functional Condition to be analyzed.

4.1. Assembly Modeling

As a result of this step, a model is defined to represent the artifact structure and the feature-based specification. The BDD shown in Figure 14 represents the artifact decomposition of the CoinageSystem in three subassemblies (DieSystem, PunchSystem and Press) and their structure.

Focusing on the DieSystem specification, Figure 14 represents composition relationships with «Part» Blocks (DieTool and DieToolHolder) and «GT_Specification» Blocks (Coaxiality and Distance_parallel_planes_specif). These «GT_Specification» Blocks are used to define the assembly relationships between parts as depicted in the IBD of Figure 15.

This diagram also represents some features of both parts that are offered as Ports for establishing the assembly relationships of the DieSystem with other Artifacts.

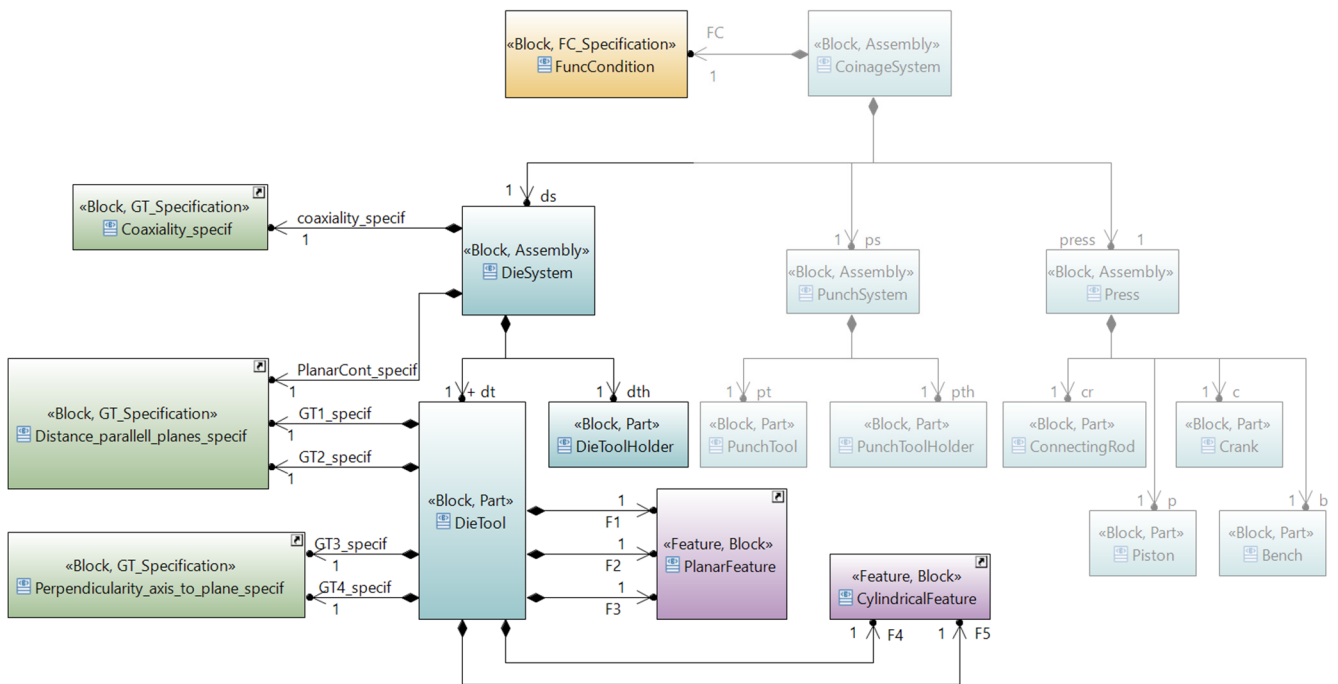


Figure 14. BDD to represent the Artifact structure of the CoinageSystem Block, focused on the elements that participate in the analysis model in next steps.

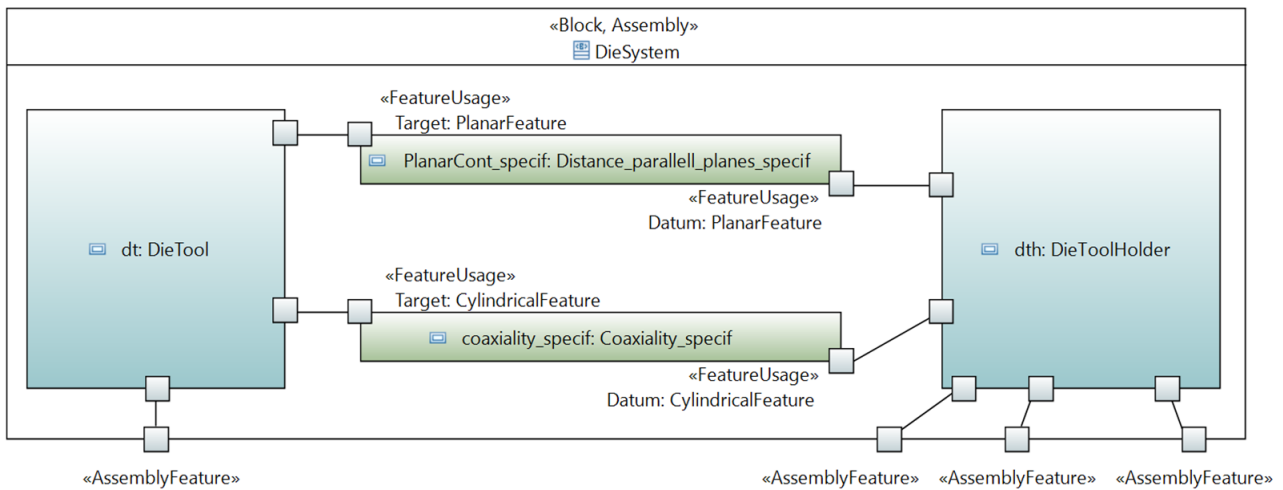


Figure 15. IBD to represent the Feature-based internal structure of the DieSystem Block.

As for the DieTool, Figure 14 specifies its composition, including «Feature» Blocks (Planar-Feature and CylindricalFeature) and «GT_Specification» Blocks (Perpendicularity_axis_to_plane_specif and Distance_parallell_planes_specif). In this case, the «GT_Specification» Blocks are supporting the specification of the part geometry. IBD of Figure 16 shows its internal structure, where some properties (features and GTs) are depicted with remarked edges to identify elements that will be relevant for the tolerance analysis proposed in the next steps. Specifically, these elements are those involved in the position specifications depicted in Figure 17, where the specifications according to standardized notation and the model elements are represented.

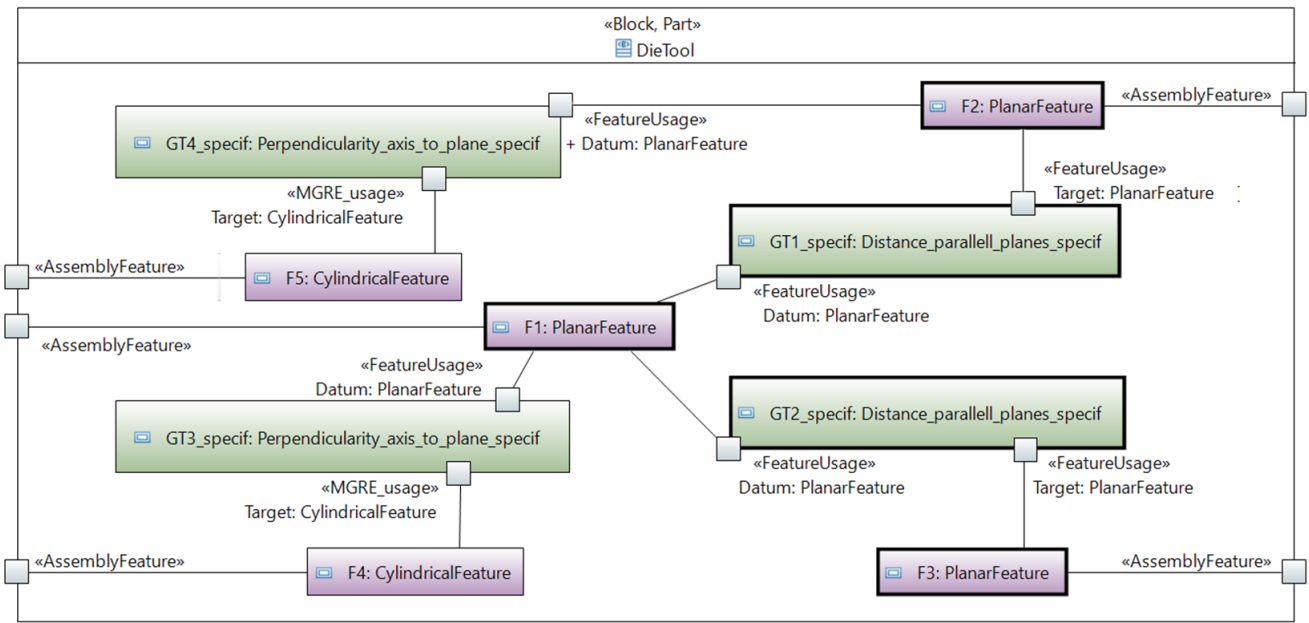


Figure 16. IBD to represent the Feature-based internal structure of the DieTool Block.

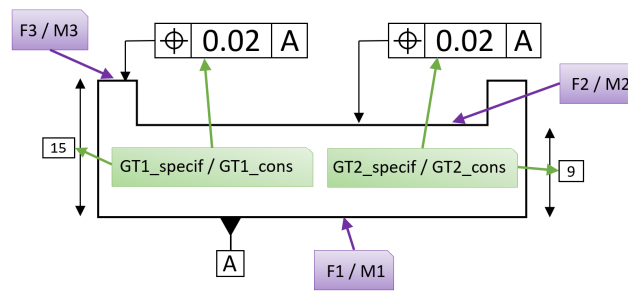


Figure 17. Drawing of the die tool part with specifications for the tolerance analysis.

4.2. TTRS-Based Modeling

As a result of the second step, new elements are added to the previously defined Artifacts to incorporate the TTRS-based representation of specified features and GTs, including mathematical definitions for these elements and relationships. The BDD of Figure 18 represents additional composition relationships to include some TTRS-based representation for Properties of the DieTool Block, specifically those elements used in the analysis developed in the next steps (parts marked with a thick border in Figure 16). Moreover, a new IBD of the DieTool Block is defined to represent its internal structure from a TTRS point of view (Figure 19).

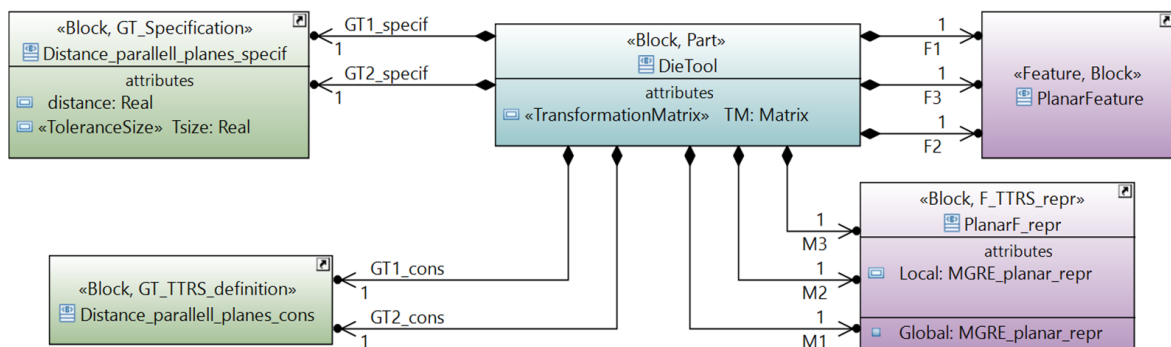


Figure 18. BDD to represent TTRS-based structure of the DieTool Block.

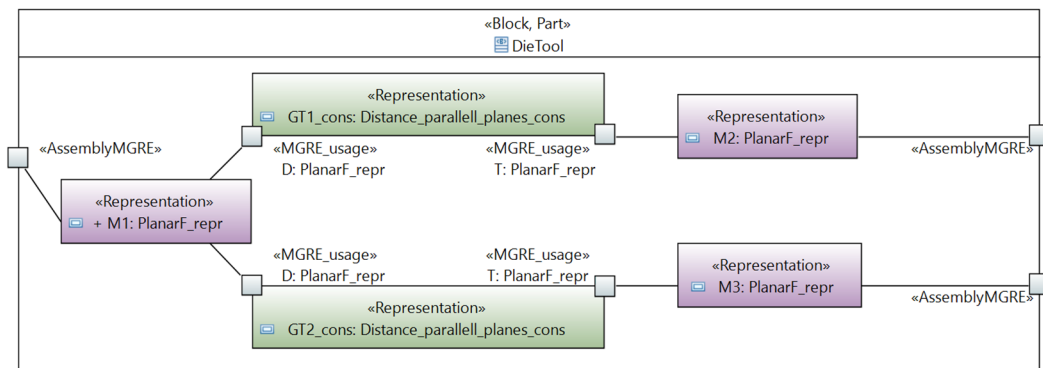


Figure 19. IBD to represent the TTRS-based internal structure of DieTool Block.

The model also contains the necessary connections between the transformation matrix of DieTool Block (TM Property) and each PlanarF_repr (M1, M2 and M3), although it is not represented graphically for the sake of clarity.

4.3. Tolerance Analysis Modeling

The third step of the methodology is exemplified by the analysis of the proposed Functional Condition that is depicted in Figure 13. To analyze this Functional Condition, a chain is built using elements from the TTRS_based representation of the PunchTool and the DieTool. In this particular case, the six MGREs (M1, M2, M3 . . .) are the nodes (Node1, Node2, Node3 . . .), and the four GTs between MGREs are the links (Link1, Link2, Link3, Link4), as shown in Figure 20a. Figure 20b depicts the chain elements and their dependencies with the elements of the TTRS-based representation. Moreover, one additional inter-part constraint (ConfigCond) is specifically added to define the assembly configuration, characterized by the contact between surfaces Node3 and Node4. Finally, the chain is closed by the Functional Condition (Figure 20c).

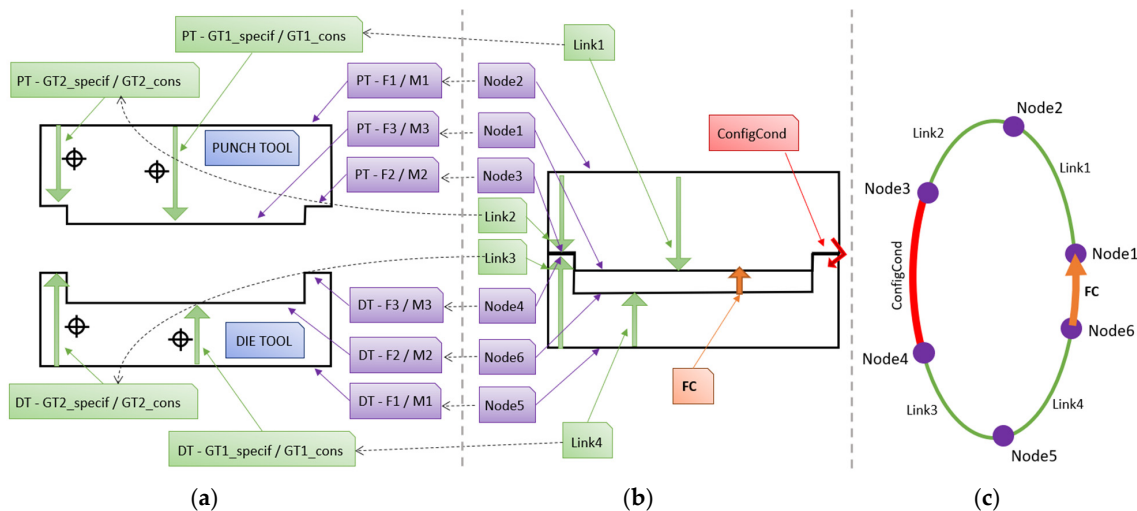


Figure 20. Graphical representation of the tolerance analysis developed: (a) elements involved in the analysis, (b) additional constraint and Functional Condition definition, (c) Functional Chain to be analyzed.

The BDD of Figure 21 represents the structure of the CoinageSystemAnalysis Block, which includes an AnalysisFC_1 Block for the analysis of the proposed Functional Condition. This AnalysisFC_1 Block is composed of a «ChainConstruction» Block (Chain_1) and a «AnalysisSpecification» Block (AnalysisSpecif_1).

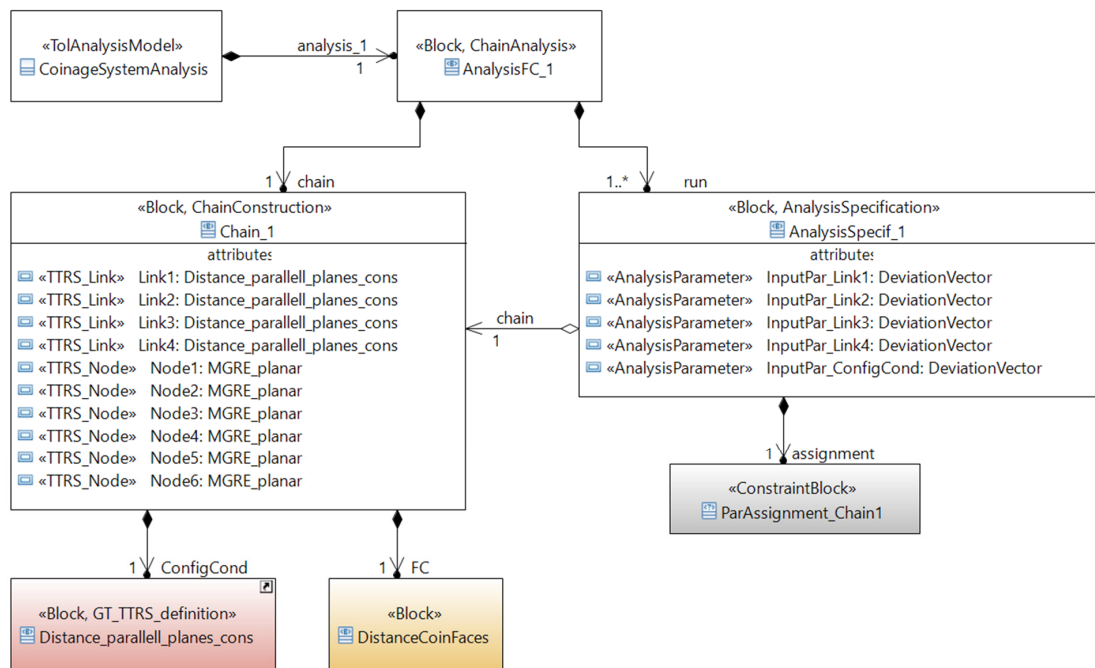


Figure 21. BDD of the general tolerance analysis model structure (“1..*” is a “one to many” multiplicity).

On the one hand, Chain_1 Block defines the chain construction, and it is composed of (a) «TTRS_Link» Properties (Link1, Link2 . . .), (b) «TTRS_Node» Properties (Node1, Node2 . . .), (c) an FC Property, typed by a «FunctionalCondition» Block (DistanceCoinFaces), and (d) ConfigCond property, added to define the assembly configuration. The internal structure of Chain_1 Block is shown in the IBD of Figure 22, where the chain is represented, showing its complete and closed definition.

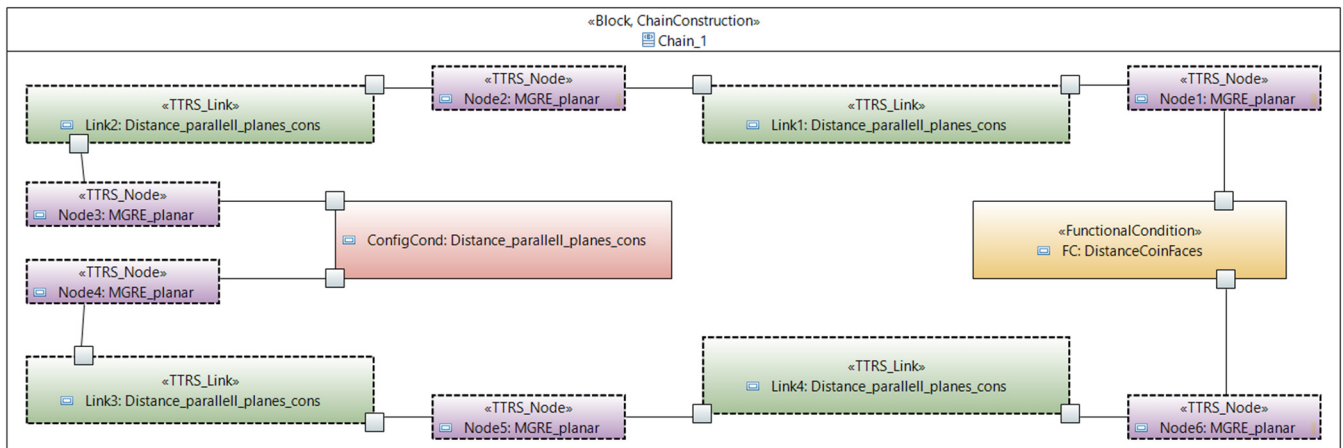


Figure 22. IBD of the Chain_1 Block.

On the other hand, the AnalysisSpecif_1 Block includes (a) the previous Chain_1 Block (shared relationship), (b) the necessary «AnalysisParameter» properties (InputPar_Link1, for example) to define the input analysis Parameters, and (c) the ParAssignment_Chain1 ConstraintBlock to assign input Parameters to Properties of the TTRS_Links. This assignment is represented in the PAR diagram in Figure 23.

The previously described model, defined using SysML4TA, supports the definition of all the necessary elements for the analysis of the identified Functional Condition, including the detailed definition of all the geometries involved in the Functional Chain and the mathematical relations that relate them. Thus, from this SysML model, model transformations can be automated to obtain the simulation model in a textual language (such as Modelica)

that allows the resolution of equations system. In these automatic transformations, the SysML profiles also play an important role in facilitating the mapping between languages, but these issues are outside the scope of this work and are topics for future research.

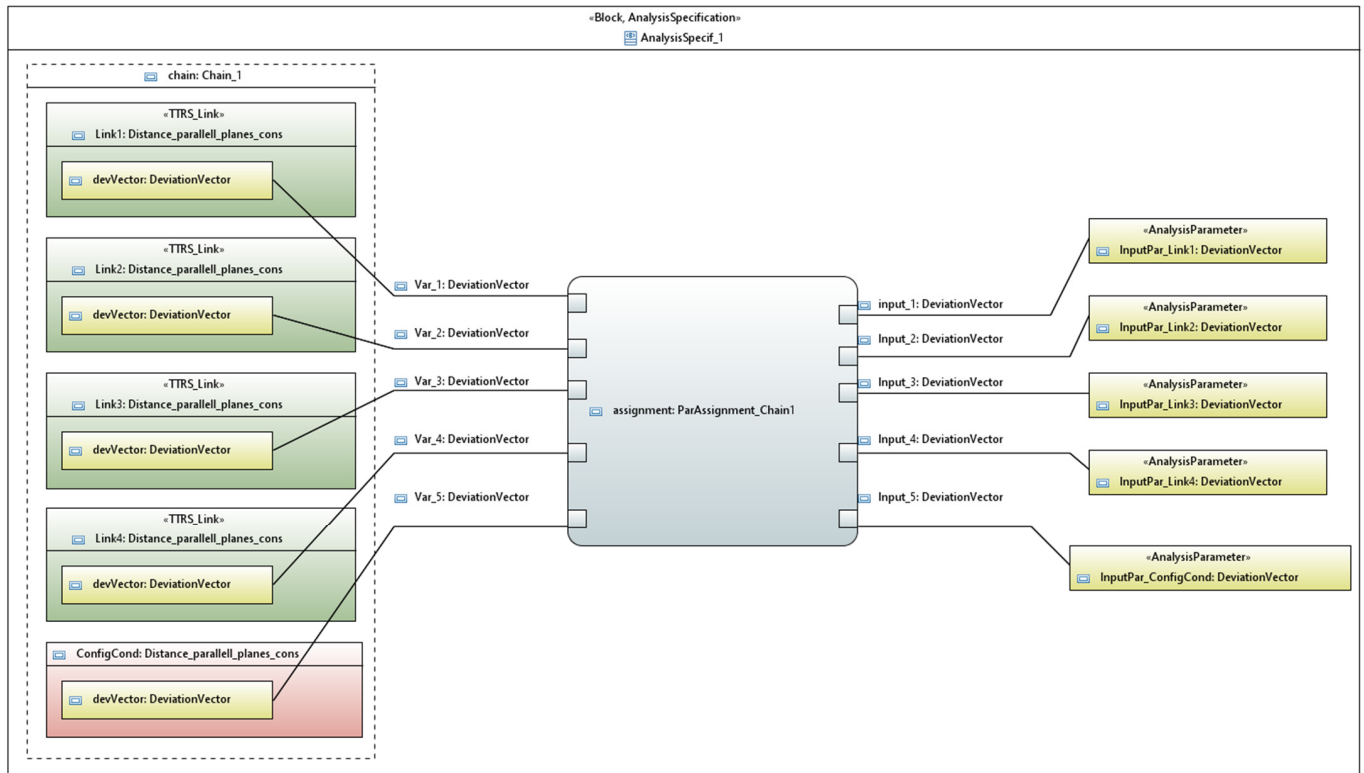


Figure 23. PAR diagram of the AnalysisSpecif_1 Block.

5. Discussion

As has been seen throughout this paper, the use of SysML in mechanical modeling is usually limited to showing the high-level functional structure of the system. Our proposed SysML central model is much more detailed, specifically in those aspects related to geometry (coordinate systems, invariance classes, MGRÉ, etc.), that are involved in system functionality. This detailed model enriches communication between experts, applications, and tools from various fields of mechanical engineering. In addition, it should be noted that the proposal supports the quantitative representation of geometric deviations (although it does not support form errors), and the mathematical equations that govern the analysis from different approaches (statistical, worst case, etc.). This is an important advantage compared to other analyzed proposals, in which SysML is used for the functional description of the system, developing qualitative analyses of the physical structure like the proposal based on C&C-M [19,20]. Even GERTRUDE profile [18], which is based on the TTRS model, does not exploit the mathematical definition of surfaces with the level of detail considered this paper.

After developing the proposal and its application to a case study, we have concluded that SysML has adequate constructive elements to define analytical models. This language includes constructions for a mathematical definition of the system and diagrams for its graphical representation from different perspectives (mainly BDDs, IBDs, and PAR diagrams in this proposal). Given the generality of SysML, the SysML4TA profile was developed as a DSML for tolerance analysis, integrating concepts from the TTRS model. Along with the definition of the proposed profile and library and their subsequent application to a case study, the validity of TTRS concepts for supporting the geometry modeling has been demonstrated, including the definition of deviated geometries by applying SDTs. Additionally, it should be noted that throughout the development of the proposal, some

difficulties have been overcome. For example, SysML associative blocks have limitations in representing non-binary relationships, so alternative modeling solutions were explored. Regarding representation through diagrams, the case study highlights the convenience of enriching formal SysML modeling with alternative graphical representations to facilitate the definition and interpretation of models related to a specific domain. Thus, the development of a specific profile could allow for the definition of additional diagrams, designed from the perspective of a particular domain. In the specific case of this proposal, it would be interesting to have diagrams closer to the sketches usually used by tolerance analysis and synthesis experts (e.g., Figure 20c), integrating different viewpoints that current SysML diagrams address separately.

Regarding the use of OCL rules for consistency assurance, this study explored the definition of OCL rules in each stereotype of the SysML4TA profile. These OCL rules restrict the use of handled concepts, detecting inconsistencies when any of these rules are not fulfilled during model validation. The developed rules, described in the appendixes, control various model aspects, such as its completeness, the use of admissible connections, or the Functional Chain correctness, among others. In order to test the proposed OCL rules, during the development of the case study, conceptual errors were intentionally introduced into the model, and the model validation processes successfully identified them. Moreover, other unintentional errors were also identified. This proves the potential of the use of OCL rules at the profile level, as well as the correct implementation and operation of these rules. However, some issues about the completeness of the analysis model could not be addressed by defining rules in the profile level, and their validation can only be supported with OCL rules defined in the system model.

Beyond consistency assurance, this proposal takes advantage of some SysML characteristics that facilitate subsequent transformations of these models to other languages. For example, the compliance with the encapsulation principle in the design of the proposed Blocks is aligned with the object orientation of analysis languages such as Modelica and the way in which Blocks are usually connected in these languages. Moreover, most of the defined SysML Blocks contain their own Constraint Blocks, in which the mathematical equations that support the behavior of the system are defined. This structure, in which each Block contains equations to define its behavior, is also typical in other languages. Besides language-specific issues, another key issue to facilitate model transformations is having available equivalent libraries developed in both languages. In this sense, the TA_Library described in this article was developed in parallel with an equivalent Modelica element library, presented in [31]. This strategy allows for establishing relationships between these constructions and simplifying the mapping between the models built with these library elements.

Finally, regarding the last of the research questions, this paper describes the procedure to be followed for the creation of the proposed models, considering three main steps: assembly specification, TTRS-based representation, and tolerance analysis modeling. However, the need to integrate these steps with the modeling tasks of the systems architect arises. In this sense, the proposed procedure is integrated into a broader framework, in which the relationships (fundamentally automated transformations) between models from various domains (and defined under various meta-models) will be explored in future works.

6. Conclusions and Future Work

In this work, SysML has been adopted as the basic system engineering language to develop a DSML for tolerance analysis, implemented as a SysML profile: SysML4TA. The main concepts of this language were depicted in different diagrams as a representation of the abstract syntax of the metamodel, but these concepts were implemented in SysML4TA as 38 stereotypes, enriched with 60 OCL rules that support the domain-specific semantics. This paper also presented an initial library version (TA_Library) that specifies a basic set of reusable and consistent constructions for the effortless creation of the tolerance analysis model for a particular system. The use of these resources (SysML4TA and TA_Library)

following the proposed procedure facilitates the tolerance analysis modeling, assuring the consistency of the model according to the domain-specific semantics of the handled concepts. A case study to show the application of the methodology and its potential to analyze the geometric variability of a single-stage manufacturing process was also presented. We hope that this contribution will help to increase the acceptance of MBSE in the field of mechanical engineering and its application in an integrated way. As mentioned, SysML can be a key language in this important task to complete the digital thread that links the function with the manufacturing.

In continuation of this research, three major work lines are planned. The first is the extension of the proposed DSML to incorporate semantics of system dynamics, which is necessary to develop consistent SysML models of hybrid simulation, where both the flow of materials and the flow of geometric variability are considered for multi-stage manufacturing processes simulation. Secondly, the automated model-to-model and model-to-text transformations will be deeply explored to create Modelica executable simulation models from SysML models and to support integration and synchronization with other models, for example, CAD models. Finally, a more ambitious work line is the specification of the concepts handled in the profile as an ontological model. This approach is compatible with the use of reasoners to support a deeper consistency assurance and knowledge inference about the domain. This additional ontological model must be also integrated with other models.

Author Contributions: Conceptualization, S.B.-N., P.R.C. and F.R.S.; methodology, S.B.-N. and F.R.S.; software, S.B.-N. and P.R.C.; validation, S.B.-N. and P.R.C.; formal analysis, S.B.-N. and F.R.S.; investigation, S.B.-N. and F.R.S.; resources, S.B.-N.; data curation, J.V.A.-N.; writing—original draft preparation, S.B.-N., P.R.C. and F.R.S.; writing—review and editing, J.V.A.-N.; visualization, S.B.-N.; supervision, P.R.C. and F.R.S.; project administration, P.R.C. and J.V.A.-N.; funding acquisition, J.V.A.-N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by “Pla de Promoció de la Investigació a l’UJI” of UNIVERSITAT JAUME I. Project title: “Modelado e Implementación del gemelo digital para Sistemas Ciber-físicos de Producción (CPPS) orientado al aseguramiento de la calidad geométrica de producto”. Reference: UJI-B2020-33.

Data Availability Statement: The data presented in this study are available on request to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Description of main ArtMPP stereotypes.

Stereotype	Extends to	Definitions and Constraints
«Artifact»	«Block»	Specification of a distinctive entity (component) in a physical system. It is an abstract stereotype whose specializations are «Part» and «Assembly».
«Part»	«Artifact» («Block»)	Specification of an atomic Artifact that cannot be decomposed in other Artifacts, so a «Part» Block cannot contain another «Artifact» («Part» or «Assembly») Block (C1). Any «Part» Block must contain at least one «Feature» or «F_TTRS_Repr» Block (C2).
«Assembly»	«Artifact» («Block»)	Specification of a composite Artifact that is composed of at least two «Artifact» («Part» or «Assembly») Blocks (C3). Any property typed by an «Artifact» Block must own at least one «AssemblyFeature» or «AssemblyMGRE» Port (C4) to support its assembly relationships with other Artifacts.

Table A2. Description of main FMPP stereotypes.

Stereotype	Extends to	Definitions and Constraints
«Feature»	«Block»	The specification of a shape portion of an Artifact. Every «Feature» Block must belong to an «Artifact» («Part» or «Assembly») Block (C6) to avoid orphaned features.

Table A2. *Cont.*

Stereotype	Extends to	Definitions and Constraints
«FC_Specification»	«Block»	Specification of a Functional Condition defining the acceptable variability limits of a target feature with respect to some datum features. A «FC_Specification» Block must own a «ToleranceSize» Property and 2 to 4 «FeatureUsage» Ports (C8).
«GT_Specification»	«Block»	Specification of a geometric tolerance defining the size of the tolerance zone that represents the limits of the acceptable variability according to a type of positional/orientational imposed relationship from some datum features to a target feature. A «GT_Specification» Block must own a «ToleranceSize» Property and 2 to 4 «FeatureUsage» Ports (C9).
«ToleranceSize»	UML4SYSML:: «Property»	Property that specifies the size of a tolerance zone (Real value), so a «ToleranceSize» Property must be owned by a «FC_Specification» or «GT_Specification» Block and be typed by a Real (C10).
«FeatureUsage»	UML4SYSML:: «Port»	Port that represents a Feature participating in a geometric tolerance specification, either as datum or target. To define the role, the «FeatureUsage» stereotype has an Attribute (“role”) typed by an Enumeration with the EnumerationLiteral (datum, target). Moreover, the «FeatureUsage» stereotype has an Attribute (“order”) typed by an Enumeration that includes EnumerationLiteral (none, FirstRef, SecondRef, ThirdRef). If the «FeatureUsage» Port is a target (role = target), no order can be defined (order = none), and if the «FeatureUsage» Port is a datum (role = datum), order must be defined (order ≠ none) (C11). A «FeatureUsage» Port is always owned by a «GT_Specification» or «FC_Specification» Block and typed by a «Feature» Block (C12).
«Assembly Feature»	UML4SYSML:: «Port»	Port that represents a Feature offered by an Artifact to participate in assembly relationships. A «Assembly Feature» Port must be typed by a «Feature» Block and be owned by an «Artifact» («Part» or «Assembly») Block (C13).

Table A3. Description of main TTRS_MPP stereotypes.

Stereotype	Extends to	Definitions and Constraints
«F_TTRS_Repr»	«Block»	A TTRS-based representation of a feature, including MGREs defined with respect to the Local and the Global reference system related by a transformation. For this reason, a «F_TTRS_Repr» Block must contain (a) a «MGRE_repr» Block, representing the local definition; (b) a behavior Port typed by a «MGRE_repr» Block, representing the global definition; (c) a ConstraintBlock that supports the transformation between Local and Global reference systems; (d) a «TM_Port» behavior Port, that enables using the transformation matrix defined in the owner «Part» Block (artifact) (C14). Any Property typed by a «F_TTRS_Repr» Block always has the «Representation» Stereotype applied (C15).
«MGRE»	«Block»	Minimal geometric reference elements for a surface according to its invariant class. An «MGRE» Block is composed of at least one «RGE» Block and at most three «RGE» Blocks (C16). A «MGRE» Block must contain as many «RGE Constraint» ConstraintBlocks as the number of RGEs minus 1 (C17). If the «MGRE» Block contains more than one «RGE» Block, each one must be connected with at least one Parameter of an «RGE Constraint» ConstraintBlock (C18). The seven types of MGRE are defined at a library level.
«MGRE_repr»	«Block»	A TTRS-based representation of a feature, including both the nominal and deviated definition. For this reason, an «MGRE_repr» Block must contain a) a «MGRE_usage» behavior Port, representing the nominal definition and b) a «MGRE_usage» behavior Port, representing the deviated definition (C19).
«MGRE_usage»	UML4SYSML:: «Port»	Behavior Port typed by a «MGRE» Block (20), differentiating its use to support nominal or deviated definition of a geometry. The «MGRE_usage» stereotype has a Boolean Attribute (isDeviated) that distinguishes between its use to define nominal (“false”) or deviated (“true”) geometries.
«RGE»	«Block»	A reduced geometric element (TTRS concept) that is used to construct an MGRE. Any «RGE» Block must be owned by a «MGRE» Block (C21). An RGE can be points, planes, or lines, but these elements are defined at the library level.

Table A3. Cont.

Stereotype	Extends to	Definitions and Constraints
«TTRS»	«Block»	Entity that represents a pair of related MGREs, defining constraints between them. A «TTRS» Block must own two Ports typed by «MGRE» Blocks, two Properties typed by «MGRE» Blocks, and at least one «RGE Constraint» ConstraintBlock (C22).
«RGE Constraint»	«ConstraintBlock»	Geometric constraint between two RGEs, corresponding to the 13 simple cases shown in Figure 2. These 13 constraints are defined at the library level. The «RGE Constraint» ConstraintBlock must own two Parameters typed by «RGE» Blocks (C23).
«Representation»	UML4SYSML:: «Property»	Property that represents another element of the model from a different viewpoint. The «Representation» stereotype has an Attribute (“representsTo”) to define the element it represents. The owner of a «Representation» Property must also own another Property whose name coincides with the value of the “representsTo” Attribute (C24).
«Assembly MGRE»	UML4SYSML:: «Port»	Port typed by a «F_TTRS_Repr» block (C25) that represents a geometry offered by an Artifact to participate in assembly relationships. An «Assembly MGRE» Port must be owned by an «Artifact» («Part» or «Assembly») Block (C26).
«TransformationMatrix»	UML4SYSML:: «Property»	Homogeneous matrix that supports the translation and rotation of geometry definitions to change the reference system considered. It is a Property that characterizes a Part, so every geometry owned by this Part must be transformed in the same way. For this reason, the owner «Part» Block must contain only one «TransformationMatrix» Property (C27).
«TM_Port»	UML4SYSML:: «Port»	Behavior Port owned by an «F_TTRS_Repr» Block that is connected with the «TransformationMatrix» Property of the Part owning the «F_TTRS_Repr» Block (C28).
«GT_TTRS_definition»	«Block»	Mathematical definition of a geometric tolerance between TTRS-based feature representations. Any Property typed by a «GT_TTRS_definition» Block always has the «Representation» stereotype applied (C29). A «GT_TTRS_definition» Block must own a «TZ_TTRS_definition» Block, 2 to 4 «TTRS_Repr_usage» Ports (C42), and the same quantity of Properties typed by «F_TTRS_Repr» Blocks as «TTRS_Repr_usage» Ports (C30). Moreover, a «GT_TTRS_definition» Block must own the same or higher quantity of Properties typed by «TTRS» Blocks than the double of the number of «TTRS_Repr_usage» Ports with the Attribute role = datum (C31) to relate the tolerance zone with the considered nominal and deviated data. A «GT_TTRS_definition» Block must own (direct or indirectly) two Properties typed by «SpecifValidation» and «TargetConstruction» ConstraintBlocks, respectively, that relate the tolerance zone with the target geometry and a «TargetDeviation» Property in which deviations for the target constructions are defined (C32).
«SpecifValidation»	«ConstraintBlock»	Validation of the specification consistency, checking that the nominal MGRE of the target coincides or is included in the nominal MGRE of the tolerance zone (constructed from the nominal MGRE of the datums). A «SpecifValidation» ConstraintBlock includes a Boolean Parameter (C33) to transfer the result of the validation.
«TargetConstruction»	«ConstraintBlock»	Construction of the deviated MGRE for the target geometry, using a «DeviationVector» Parameter (C34) according to the limits imposed by the tolerance zone MGRE and its size.
«TZ_TTRS_definition»	«Block»	A geometrical space used to control/define the position/orientation of a target geometry. This geometrical space is characterized by a size, specified in a geometric tolerance specification, so a «TZ_TTRS_definition» Block owns an AdjunctProperty that points to the «ToleranceSize» Property of a Property typed by a «GT_Specification» Block (35). A «TZ_definition» Block also owns two «MGRE» Blocks, one of them with the Attribute “isNominal = true” and another with the Attribute “isNominal = false” (C36).

Table A3. Cont.

Stereotype	Extends to	Definitions and Constraints
«TTRS_Repr_usage»	UML4SYSML:: «Port»	TTRS-based representation used as a reference (datum) or target element in the definition of a geometric tolerance. Any «TTRS_Repr_usage» Port is always owned by a «GT_Specification» or «FC_Specification» Block and typed by an «F_TTRS_Repr» Block (C37). The «TTRS_Repr_usage» stereotype has an Attribute (“role”) typed by an Enumeration with EnumerationLiterals (datum, target), and an Attribute (“order”) typed by an Enumeration with EnumerationLiterals (none, FirstRef, SecondRef, ThirdRef). If the «TTRS_Repr_usage» Port is a target (role = target), no order can be defined (order = none), and if the «TTRS_Repr_usage» Port is a datum (role = datum), order must be defined (order ≠ none) (C38).
«TargetDeviation»	UML4SYSML:: «Property»	Property that expresses the displacements (deviations) in the variant DoFs of the target MGRE. Any «TargetDeviation» Property must be owned by a «GT_TTRS_definition» or «GT_TTRS_deviation» Block and must be typed by a «DeviationVector» DataType (C39).
«DeviationVector»	UML4SYSML:: «DataType»	DataType that is defined to express the small displacements of a deviated MGRE with respect to the nominal position. All the DataValues considered must be typed by a Real primitive (C40).

Table A4. Description of main TAPP stereotypes.

Stereotype	Extends to	Definition and Constraints
«TolAnalysisModel»	«Block»	Block that includes analysis of different Functional Chains. Any «TolAnalysisModel» Block is composed of at least one «Chain Analysis» Blocks (C41). A «TolAnalysisModel» Block must aggregate the «Artifact» («Part» or «Assembly») Block to be analyzed (C42).
«ChainAnalysis»	«Block»	Block that contains the tolerance analysis of a Functional Chain. It must contain a «ChainConstruction» Block and at least one «AnalysisSpecification» Block (43).
«ChainConstruction»	«Block»	The definition of the Functional Chain to be analyzed, including the Functional Condition, the links and nodes involved, and other additional constraints to define the configuration, if necessary. Any «ChainConstruction» Block must include a «Functional Condition» Block, at least two «TTRS_Link» AdjunctProperties, and at least three «TTRS_Node» AdjunctProperties (C44). All owned «TTRS_Link» or «TTRS_Node» Blocks must point to constructions with a common parent «Artifact» Block (C45). Moreover, it is necessary to ensure that the chain is closed (C46).
«Analysis Specification»	«Block»	Specification of an analysis for a Functional Chain, where input Parameters are defined according to the tolerance stacking method. To establish the input Parameters, an «AnalysisSpecification» Block must own at least one «AnalysisParameter» Property and a «ParamAssignment» ConstraintBlock (C47) to assign those Parameters to Properties of each link. Moreover, an «AnalysisSpecification» Block must own an «FC_Value» Property and an «FC_Requirement» AdjunctProperty, as well as an «FC_Compliance» ConstraintBlock in which both values are compared (C48).
«Functional Condition»	UML4SYSML:: «Property»	A Functional Condition under analysis, generally defined in the specification model (feature-based representation). A «FunctionalCondition» Property must have also applied the «Representation» stereotype (C49). Structurally, a Functional Condition is equivalent to a geometric tolerance where deviations are not defined, but calculated by the rest of the Functional Chain, so a «FunctionalCondition» Property must be typed by a «GT_TTRS_deviation» Block (C50).
«FC_Value»	«AdjunctProperty»	AdjunctProperty owned by a «ChainAnalysis» Block (C51) to store the accumulated deviation of the Functional Condition. An «FC_Value» AdjunctProperty must point to a property of the «TZ_TTRS_definition» Block owned by the «Functional Condition» Block (C52).

Table A4. Cont.

Stereotype	Extends to	Definition and Constraints
«FC_Requirement»	«AdjunctProperty»	AdjunctProperty owned by a «ChainAnalysis» Block (C53) to store the limits imposed by the Functional Condition specification, so the «FC_Requirement» AdjunctProperty must point to a «ToleranceSize» Property of an «FC_Specification» Block (C54).
«FC_Compliance»	«ConstraintBlock»	ConstraintBlock owned by a «ChainAnalysis» Block (C55) to prove with an inequation if the calculated deviation is lower than the requirement. One Parameter of the «FC_Compliance» ConstraintBlock must be connected with an «FC_Value» AdjunctProperty and another to the «FC_Requirement» AdjunctProperty (C56).
«TTRS_Link»	«AdjunctProperty»	A link of the Functional Chain, that is, one of the TTRS-based definitions of geometric tolerances that participates in the constructed chain. A «TTRS_Link» AdjunctProperty must be owned by a «ChainAnalysis» Block, and its Principal Attribute must point to a «GT_TTRS_deviation» Block (C57).
«TTRS_Node»	«AdjunctProperty»	A node of the Functional Chain, that is, one of the TTRS-based feature representations that participates in the constructed chain. A «TTRS_Node» AdjunctProperty must be owned by a «ChainAnalysis» Block and point to an «F_TTRS_Repr» Block (C58). Each «TTRS_Node» AdjunctProperty must be connected with a «TTRS_Link» AdjunctProperty or a «Functional Condition» Block through their Ports (C59).
«AnalysisParameter»	UML4SYSML:: «Property»	Property owned by a «ChainAnalysis» Block and typed by a to a «DeviationVector» DataType (C60) to state the values of deviations considered depending on the tolerance stacking method (worst case, statistical, Monte Carlo simulation, etc.).

Appendix B

Table A5. OCL expressions corresponding to some described rules.

Rule	OCL Expression
C1	self.base_Class.allAttributes()->select(a a.type.oclIsKindOf(UML::Class)).type.oclAsType(UML::Class).getAppliedStereotypes().allParents()->select(b b.name = 'Artifact')->isEmpty()
C4	self.base_Class.allAttributes()->select(a a.type.oclIsKindOf(UML::Class)).type.oclAsType(UML::Class)->select(b b.getAppliedStereotypes().allParents()->select(c c.name = 'Artifact')->notEmpty())->forAll(d d.ownedPort.getAppliedStereotypes()->select(e e.name = 'AssemblyFeature' or e.name = 'AssemblyMGRE')->notEmpty())
C46	self.base_Class.allAttributes()->select(b b.type.oclIsKindOf(UML::Class))->select(c c.getAppliedStereotypes()->select(d d.name = TTRS_)->notEmpty()) ->forAll(node self.base_Class.allFeatures()->select(f f.oclIsKindOf(UML::Connector)).oclAsType(UML::Connector).ownedElement->select(g g.oclIsKindOf(UML::ConnectorEnd)).oclAsType(ConnectorEnd).role->select(h h.oclType() = UML::Property)->select(i i = node)->size() = 2) and self.base_Class.allAttributes()->select(j j.type.oclIsKindOf(UML::Class))->select(k k.getAppliedStereotypes()->select(l l.name = 'TTRS_Link')->notEmpty())->forAll(link self.base_Class.allFeatures()->select(n n.oclIsKindOf(UML::Connector)).oclAsType(UML::Connector).ownedElement->select(o o.oclIsKindOf(UML::ConnectorEnd)).oclAsType(UML::ConnectorEnd).partWithPort->select(p p = link)->size() = 2) and let C:Integer = self.base_Class.allFeatures()->select(a a.oclIsKindOf(UML::Connector))->size() in C>0 and let N:Integer = self.base_Class.allAttributes()->select(b b.type.oclIsKindOf(UML::Class)).getAppliedStereotypes()->select(c c.name = 'TTRS_Node')->size() in C=2*N and let L:Integer = self.base_Class.allAttributes()->select(b b.type.oclIsKindOf(UML::Class)).getAppliedStereotypes()->select(c c.name='TTRS_Link')->size() in N = L

References

1. Akundi, A.; Lopez, V. A Review on Application of Model Based Systems Engineering to Manufacturing and Production Engineering Systems. *Procedia Comput. Sci.* **2021**, *185*, 101–108. [CrossRef]
2. Batteux, M.; Prosvirnova, T.; Rauzy, A. Model Synchronization: A Formal Framework for the Management of Heterogeneous Models. In *Model-Based Safety and Assessment*; Papadopoulos, Y., Aslansefat, K., Katsaros, P., Bozzano, M., Eds.; Springer: Cham, Switzerland, 2019; Volume 11842. [CrossRef]
3. Torres, W.; van den Brand, M.G.J.; Serebrenik, A. A systematic literature review of cross-domain model consistency checking by model management tools. *Softw. Syst. Model.* **2021**, *20*, 897–916. [CrossRef]
4. Feldmann, S.; Kernschmidt, K.; Wimmer, M.; Vogel-Heuser, B. Managing inter-model inconsistencies in model-based systems engineering: Application in automated production systems engineering. *J. Syst. Softw.* **2019**, *153*, 105–134. [CrossRef]
5. Qamar, A.; Paredis, C.J.J.; Wikander, J.; Doring, C. Dependency Modeling and Model Management in Mechatronic Design. *J. Comput. Inf. Sci. Eng.* **2012**, *12*, 041009. [CrossRef]
6. Höpfner, G.; Jacobs, G.; Zerwas, T.; Drave, I.; Berroth, J.; Guist, C.; Rumpe, B.; Kohl, J. Model-Based Design Workflows for Cyber-Physical Systems Applied to an Electric-Mechanical Coolant Pump. In Proceedings of the 19th Drive Train Technology Conference (ATK 2021), Aachen, Germany, 9–11 March 2021. [CrossRef]
7. Wagner, H.; Zuccaro, C. Collaboration between System Architect and Simulation Expert. In Proceedings of the IEEE International Symposium on Systems Engineering, Vienna, Austria, 24–26 October 2022. [CrossRef]
8. Benavent Nacher, S.; Rosado Castellano, P.; Romero Subirón, F.; Abellán-Nebot, J.V. Multidomain Simulation Model for Analysis of Geometric Variation and Productivity in Multi-Stage Assembly Systems. *Appl. Sci.* **2020**, *10*, 6606. [CrossRef]
9. Abellan-Nebot, J.V.; Liu, J.; Romero, F. Design of multi-station manufacturing processes by integrating the stream-of-variation model and shop-floor data. *J. Manuf. Syst.* **2011**, *30*, 70–82. [CrossRef]
10. OMG Systems Modeling Language (SysML) v. 1.6. Available online: <https://sysml.org/.res/docs/specs/OMGSysML-v1.6-19-11-01.pdf> (accessed on 13 January 2023).
11. OMG Object Constraint Language (OCL), v. 2.4. Available online: <https://www.omg.org/spec/OCL/> (accessed on 13 January 2023).
12. Moeser, G. Integration of a visually supported Function-Embodiment-Synthesis into the Concepts of Model Based Systems Engineering (MBSE). In Proceedings of the 3rd International Spring School on Systems Engineering, Paderborn, Germany, 4–8 May 2015.
13. Jagla, P.; Jacobs, G.; Siebrecht, J.; Wischmann, S.; Sprehe, J. Using SysML to Support Impact Analysis on Structural Dynamics Simulation Models. *Procedia CIRP* **2021**, *100*, 91–96. [CrossRef]
14. Brahmi, R.; Hammadi, M.; Aifaoui, N.; Choley, J.Y. Interoperability of CAD models and SysML specifications for the automated checking of design requirements. *Procedia CIRP* **2021**, *100*, 259–264. [CrossRef]
15. Berriche, A.; Mhenni, F.; Mlika, A.; Choley, J.-Y. Towards Model Synchronization for Consistency Management of Mechatronic Systems. *Appl. Sci.* **2020**, *10*, 3577. [CrossRef]
16. Kernschmidt, K.; Vogel-Heuser, B. An interdisciplinary SysML based modeling approach for analyzing change influences in production plants to support the engineering. In Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE), Madison, WI, USA, 17–20 August 2013. [CrossRef]
17. Drave, I.; Rumpe, B.; Wortmann, A.; Berroth, J.; Hoepfner, G.; Jacobs, G.; Spuetz, K.; Zerwas, T.; Guist, C.; Kohl, J. Modeling mechanical functional architectures in SysML. In Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, Virtual Event, Canada, 16–23 October 2020. [CrossRef]
18. Barbedienne, R.; Penas, O.; Choley, J.Y.; Rivière, A.; Warniez, A.; Della Monica, F. Introduction of geometrical constraints modeling in SysML for mechatronic design. 10th France-Japan Congress. In Proceedings of the 10th France-Japan/ 8th Europe-Asia Congress on Mechatronics (MECATRONICS2014-Tokyo), Tokyo, Japan, 27–30 November 2014. [CrossRef]
19. Albers, A.; Zingel, C. Interdisciplinary Systems Modeling using the Contact & Channel-Model for SysML. In Proceedings of the 18th International Conference on Engineering Design (ICED 11), Lyngby/Copenhagen, Denmark, 15–19 August 2011.
20. Albert, A.; Zingel, C. Extending SysML for Engineering Designers by Integration of the Contact & Channel—Approach (C&C2-A) for Function-Based Modeling of Technical Systems. *Procedia Comput. Sci.* **2013**, *16*, 353–362. [CrossRef]
21. Fenves, S.J.; Foufou, S.; Bock, C.; Sriram, R.D. CPM: A core model for product data. *JCISE* **2008**, *5*, 014501. [CrossRef]
22. Baysal, M.M.; Roy, U.; Sudarsan, R.; Sriram, R.D.; Lyons, K.W. The Open Assembly Model for the Exchange of Assembly and Tolerance Information: Overview and Example. In Proceedings of the DETC, Salt Lake City, UT, USA, 28 September 2004. [CrossRef]
23. Baysal, M.M. Functional and Behavioral Product Information Representation and Consistency Validation for Collaboration in Product Lifecycle Activities. Doctoral Thesis, Syracuse University, Syracuse, NY, USA, 2012.
24. Grauberger, P.; Wessels, H.; Gladysz, B.; Bursac, N.; Matthiesen, S.; Albers, A. The contact and channel approach—20 years of application experience in product engineering. *J. Eng. Des.* **2020**, *31*, 241–265. [CrossRef]
25. Riveire, A.; Clément, A.; Valade, C. The TTRS: 13 oriented constraints for dimensioning, tolerancing and inspection. *Adv. Math. Appl. Sci.* **1997**, *45*, 24–42. [CrossRef]
26. Peng, H.; Peng, Z.; Zhou, Z. Manufacturing variation modeling and process evaluation based on small displacement torsors and functional tolerance requirements. *J. Adv. Mech. Des. Syst. Manuf.* **2021**, *15*, 3. [CrossRef]

27. Cao, Y.; Liu, Y.; Mao, J.; Yang, J. 3DTS: A 3D tolerancing system based on mathematical definition. *J. Zhejiang Univ. Sci.* **2006**, *7*, 1810–1818. [[CrossRef](#)]
28. ASME. *Dimensioning and Tolerancing Y14.5—2018*; ASME: New York, NY, USA, 2018; pp. 1–344. ISBN 9780791872192.
29. Eclipse Papyrus Modeling Environment. Available online: <https://www.eclipse.org/papyrus/> (accessed on 16 January 2023).
30. Grauberger, P.; Goetz, S.; Schleich, B.; Gwosch, T.; Matthiesen, S.; Wartzack, S. A Conceptual Model Combination for the Unification of Design and Tolerancing in Robust Design. In Proceedings of the Design Society: DESIGN Conference, Cavtat, Croatia, 26–29 October 2020. [[CrossRef](#)]
31. Aguilera, D.; Rosado, P.; Benavent, S. A Modelica library to simulate geometrical and dimensional deviations in process assemblies. In Proceedings of the MESIC 2023, Sevilla, Spain, 28–30 June 2023. Abstract accepted by the Scientific Committee.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.