Modelling Geometrical Tolerances with Intervals Using ISO-Standard STEP

Eva Dyllong, Wolfram Luther, and Holger Traczinski
Institute of Computer Science and Interactive Systems
Faculty of Engineering
Gerhard-Mercator-University of Duisburg
D-47048 Duisburg, Lotharstr. 65, Germany
{luther,dyllong,traczinski}@informatik.uni-duisburg.de

A development of complex mechatronic systems requires a multitude of decentralised designing and processing systems. The various software and file formats which are involved in the development process need a quick and smooth data exchange via standardised data models and interfaces. The ISO 10303 norm STEP (STandard for the Exchange of Product model data) has been developed by the International Organisation for Standardisation (ISO) within the Technical Committee 184 "Industrial Automation System & Integration". STEP provides a basic approach for an unambiguous representation of computer interpretable product information throughout the whole life-cycle of a product [1]. There are three key components of STEP: the EXPRESS data specification language is used to define data structures, entities, attributes, relationships and constraints together with EXPRESS-G, which is intended as a graphic representation of data models [10]. The Implementation Forms provide a physical file format for data that conforms to any EXPRESS schema and a Standard Data Access Interface (SDAI) to databases and applications. The Standard Data Models include integrated resources for almost universal applicability and specific models fulfilling the requirements of a particular industrial application. Around the kernel, the Application Protocols of models are defined, based on the integrated resources and implementation methods.

Thus, the ISO-Standard STEP can be interpreted as a construction kit with application protocols, which use predefined elementary construction kits under specific rules and norm methods. An application protocol consists of three parts. The first part is the application activity model (AAM), which describes the functionality and life-cycle of a product using the SADT (Structure Analysis and Design Technique) method. The second part contains the application reference model (ARM). In this part the formal descriptions of the objects take place which are identified in AAM using the description language EXPRESS or EXPRESS-G. The Application Interpreted Model (AlM) stands at the end of the application protocol and concerns the mapping of the ARM-model into an integrated resource model.

In 1997 a new project, MechaSTEP, was initiated within the scope of ISO 10303 focusing on a neutral data format for the data exchange between mechatronic systems.

This abstract presents results in the field of the requirements analysis on ISO-Standard 10303 STEP for interval arithmetic purposes. We are mainly concerned with Part 42 of the integrated resources, which is entitled "Geometrical and Topological Representation" and with Part 47, which deals with shape variation tolerances.

Part 42 includes construction specifications for the geometrical and topological representations of an object. Furthermore, it contains requisites for an explicit representation of an object model. Geometric description alternatives are the volume based, the surface based and the wireframe based designs. Special attention is paid to Faceted Boundary REPresentation (FBREP) models belonging to the first alternative together with Manifold solid Boundary REPresentation (MBREP) and Constructive Solid Geometry (CSG) models.

Recently, there is a growing interest in researching methods for validation and verification in applications. For example, path-planning, localization and tracking of a mobile robot using techniques of scientific computation are described in [6]. In a recent project funded by the German Research Council (DFG) we implemented efficient and accurate algorithms for distance calculation between a flexible robot and a target or obstacles in the complex environment and for the resulting contact problem [7]. Robust solutions to these problems are also used in the collision-free path planning if a given end-effector is moving amid a collection of known obstacles from an initial to a desired final position. For simulation purposes, the obstacles are taken to be a collection of polyhedral surfaces. Accurate floating-point algorithms have been implemented based on suitable projections and using controlled rounding and the precise dot product whereby verified error-bounds are ensured [2, 3]. If the end-effector or the sensor is taken to be a single moving point, an efficient distance algorithm applicable to non-convex polyhedral surfaces is to be used [4]. In [9] is described, how interval methods can help a moving robot with uncertain sensors to avoid collisions with obstacles.

To allow a data transfer to mechatronic modelling and simulation software, the representation of polyhedra using the FBREP models of the STEP-standard has been highlighted. Geometric elements like points, curves, and surfaces have been described and topological relationships between them identified: the vertex points of a polyhedron in a Cartesian coordinate system as geometrical and the position of these points on its faces as topological input data. This information is given in a simple ASCII-file.

We implemented an interface which transforms this file used in our algorithms into a STEP-based one and vice-versa using Part 21 of the implementation methods in STEP for clear text encoding of the exchange structure. The output file consists of two parts: the header and the data section. The header section includes entity declarations and the data section the descriptions of the given input data using the EXPRESS language. The latter is divided into three subparts: the first for the description of the boundary of each face of the polyhedron with specification of the Cartesian coordinates, the second for the description of the faceted surface and the third for the description of the polyhedron as a closed shell. In general, STEP-files are very large: a more complex geometrical object will require several hundred lines with many pieces of repeated information. Thus, automatic generation is necessary.

In the next version, the interface should also handle cylindrical, conical, spherical, B-Spline or offset surfaces as a part of the integrated resources in Part 42.

STEP Part 47 is about shape variation tolerances. Information about tolerances is important for product definition. Many interface definitions consider tolerances only in design information like the Initial Graphics Exchange Specification (IGES), an ANSI-standard for exchanging geometric design information between CAD systems or the corresponding French national Standard d'échange et de transfert SET. The disadvantage here is that these dimensioning and tolerancing specifications are not supported in current Computer-Aided Engineering and Manufacturing systems. STEP wants to remedy this situation and integrates various single-feature and related-feature tolerances. A tolerance in STEP can either be a tolerance for sizes or for geometry. To describe tolerances for dimensions, uncertainties in lengths or angles are stored in intervals with lower and upper bounds. A range of acceptable values, also known as 'limits and fits', may be selected from a standard catalogue of acceptable ranges. Additionally, a tolerance can be defined by only one bound and by significant digits. Finally, a nominal value may have no tolerance bounds associated with it.

Geometric tolerances address the acceptable deviation of the form of a manufactured object and are usually expressed as an area or volume in which the realised form must lie [11]. Tolerances in geometry mainly concern specifications about orientation (perpendicularity, parallelism, angularity), location (concentricity, symmetry) or shape (flatness, straightness, cylindricity) [12]. In this paper, a general STEP-based dimensioning and tolerancing data model is developed and implemented.

Tolerances of measures and geometry are suitable for interval calculus in two directions. In modern modelling systems like MOBILE extensions for interval arithmetic have been added [8]. In [5] a new kinetostatic transmission element was modelled with interval arithmetic. Thus, not only uncertainties in the length of the arms of multibody systems or uncertain weights at the end-effector of a manipulator can be modelled in simulation systems but also new types of (geometrical) elements. For a standardised description of such a modelled system both types of tolerances in measure and geometry are needed. Thus far the STEP concept is basically right. Although STEP offers the possibility of storing uncertain information in intervals this information is linked with the specific objects of a model. Here, a more general usage of intervals would be desirable. For example, a complex modelled object which consists of vertices, edges, and faces is stored as an instance with exact values in STEP. Additionally, tolerances can be introduced at one point of the description, such as at a line segment. Although these tolerances affect further elements of the object, for instance a vertex point on the line, these dependencies are not stored in a STEP-file. However, they can be worked out from other information. Thus, the ability to store information about uncertainty in all parts of a STEP-file would be helpful. For this purpose a basic data type for intervals is needed.

To include self- and cross-reference tolerances, Tsai et al. [12] introduced the notion of a tolerance network which represents shape features as nodes and geometric tolerance specifications as arcs connecting nodes. They construct these networks for single pieces and, by means of recursive algorithms, for assembled products as well. The corresponding STEP product data file implements the tolerance network with appropriate back and cross references.

The other way to use interval calculus in building tolerance dependencies is by checking the consistency of given tolerances with interval arithmetic. Interval arithmetic provides a tool with which to calculate the guaranteed enclosures of computational results and enables us to determine the enclosure of a value at one part of a STEP description resulting from tolerance information in other parts of the file. If the STEP tolerance value for this part does not fit in the enclosure, the whole STEP-file is not consistent. Otherwise, the result of such a computation should be stored within the STEP-file. If there are different ways to calculate an enclosure for the same value, the intersection of the intervals should be chosen to get tighter enclosures. At this point, another reason arises for a basic interval data type.

To illustrate the benefit of interval arithmetic in tolerancing data models, we checked the complete "geometric dimensions and dimensional tolerance example" described in [11] and proved its consistency by transferring the STEP-data into the extended MOBILE modelling system. This example covers the representation of all the dimensional tolerances supported, including plus-or-minus deviations, maxima, minima and nominal dimensions, limits and fits, and significant digits.

References

[1] R. Anderl, and D. Trippner, *STEP: Standard for the Exchange of Product Model Data*, Teubner, Stuttgart, 2000.

[2] E. Dyllong, W. Luther, and W. Otten, "An accurate distance-calculation algorithm for convex polyhedra", *Reliable Computing*, 1999, Vol. 3, No. 5, pp. 241–254.

[3] E. Dyllong and W. Luther, "An accurate computation of the distance between a point and a polyhedron", Vol. 80 of *GAMM 99 Annual Meeting*, Metz, France, April 12–16, Wiley-VCH, Berlin, 2000, pp. S771–S772.

[4] E. Dyllong and W. Luther, "Distance-Calculation Between a Point and a NURBS Surface", In: P.-J. Laurent, P. Sablonnire, L. L. Schumaker (eds.), *Curve and Surface Design – Saint Malo, 1999*, Vanderbilt University Press, Nashville TN, 2000, pp. 55–62.

[5] C. Hörsken and H. Traczinski, "Modeling of multibody systems with interval arithmetic", In: W. Krämer and J. Wolff von Gudenberg (eds.), *Scientific Computing, Validated Numerics, Interval Methods, Proc. of SCAN 2000*, 2001, pp. 317–328.

[6] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter, *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer, Berlin, 2001.

[7] W. Luther, E. Dyllong, D. Fausten, W. Otten, and H. Traczinski, "Numerical verification and validation of kinematics and dynamical models for flexible robots in complex environments", In: U. Kulisch, R. Lohner, A. Facius (eds.), *Perspectives on Enclosure Methods*, Springer, Wien, 2001, pp. 181–199.

[8] W. Luther and H. Traczinski, "Error Propagation Control in MOBILE: Extended Basic Mathematical Objects and Kinetostatic Transmission Elements", In: A. Kecskeméthy, M. Schneider, and C. Woernle (eds.), *Advances in Multibody Systems and Mechatronics*, 1999, pp. 267–276.

[9] D. Morales, and T. C. Son, "Interval Methods in Robot Navigation", *Reliable Computing*, 1998, Vol. 4, No. 1, pp. 55–61.

[10] D. Schenck and P. Wilson, *Information Modeling: The EXPRESS Way*, Oxford University Press, New York, 1994.

[11] M. Strub, M. Hauser, and T. Hendrix, *Recommended Practices for Dimensions and Dimensional Tolerances*, April 18, 2000. `http://www.cax-if.org/public/Rec_Prac_DDT_v2.pdf`

[12] J.-C. Tsai, T.-C. Chuang, T.-C., and D.-N. Guo, "Development of a Step-based Dimensioning and Tolerancing Data Model", *Proc. Natl. Sci. Counc. ROC(A)*, 1998, Vol. 22, No. 6, pp. 831–840. `http://nr.stic.gov.tw/ejournal/ProceedingA/v22n6/831-840.pdf`