Computer-Aided Design 46 (2014) 148–159

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Geometric interoperability via queries

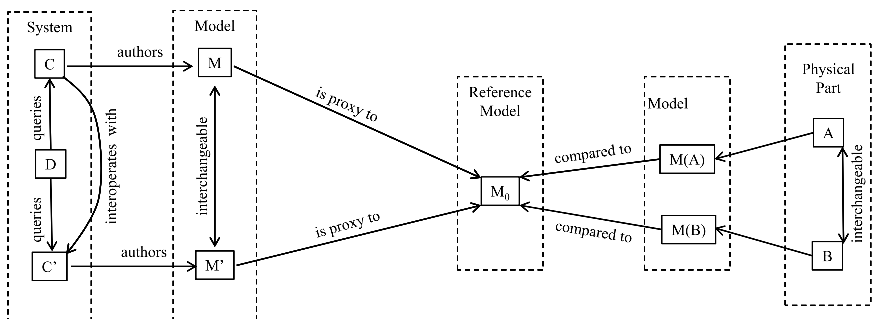
Christoph Hoffmann [a], Vadim Shapiro [b,*], Vijay Srinivasan [c]

[a] Purdue University, United States
[b] University of Wisconsin, United States
[c] National Institute of Standards and Technology, United States

HIGHLIGHTS

- Queries solve interoperability problems that are unsolved by a data-centric approach.
- Interoperability, interchangeability, and integration use a semantic reference model.
- A hierarchy solves interoperation in design and manufacturing of incidence structures.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Interoperability
Interchangeability
Integration
Geometric queries
Data exchange
Shape evaluation/comprehension

ABSTRACT

The problem of geometric (model and system) interoperability is conceptualized as a non-trivial generalization of the problem of part interchangeability in mechanical assemblies. Interoperability subsumes the problems of geometric model quality, exchange, and interchangeability, as well as system integration. Until now, most of the interoperability proposals have been data-centric. Instead, we advocate a query-centric approach that can deliver interoperable solutions to many common geometric tasks in computer aided design and manufacturing, including model acquisition and exchange, metrology, and computer aided design/analysis integration.

1. Introduction

There is a bewildering variety of shape representations in use in computer-aided design and manufacturing. From point clouds and tessellated shape representation to NURBS, from net shape to fully parameterized, different stages of the process chain use different representations. The various representations have been evolved and refined so as to respond to the specific needs of the process using them, such as design, analysis, and particular manufacturing processes, to name a few. Moreover, the representations differ in information content, as is appropriate for the task for which they are employed [1].

If digital-based manufacturing is to realize its full potential, the various representations have to be integrated and, acknowledging practice, nominally equivalent representations must be usable interchangeably and should interoperate. This is not a novel insight: there has been substantial work that attempts to reach this goal by seeking to standardize and translate models between different systems that have created them in the various representations [2]. Unfortunately, the translation approach tends to fail in certain instances owing to mathematical reasons of representability, information content, and interpretation. As a specific example, recall that shape representations such as NURBS cannot accurately represent trimmed patches. Consequently, a CAD system has to interpret face extent and the exact location of edges and vertices. Those determinations are made by algorithms that balance what

* Corresponding author.
E-mail addresses: cmh@purdue.edu (C. Hoffmann), vshapiro@engr.wisc.edu (V. Shapiro), vijay.srinivasan@nist.gov (V. Srinivasan).

0010-4485/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cad.2013.08.027

in principle amounts to uncertain or contradictory results. When such representations are translated, the needed compromises cannot be well supported, if at all. *We therefore argue that the authoring system do the needed interpretations, and that model interchangeability be based on querying models instead of translating them.*

Note that, even if a translation approach worked, it would not even begin to address the larger problem of interoperability which also requires the ability to integrate different computations and applications, as well as to communicate between systems and representations that are often based on different mathematical assumptions.

1.1. Queries, evaluation and comprehension

Despite decades of serious work the model translation approach contends with stubborn difficulties that remain to be solved fully. Therefore, we propose queries as an alternative. When system D requires data from a model M authored in system C, then D should acquire that information through a series of queries, addressed to system C about M. The queries depend on the nature of the task system D is carrying out, for which information about M is needed. This interaction between the two systems suggests a form of model interchangeability: instead of querying C about M, an equivalent model M', authored in system C', can be queried by D. In this scenario the notion of model interchangeability is relativized, restricting the domain of interchangeability to the information that is queried.

The notion of interchangeability hinges on equivalence of M and M' which applies only in the context of the application D requires information for. This context is determined by a semantic reference model appropriate to the domain of interest.

Note that model M could very well encode other information not relevant to D's task. Moreover, such interchangeability arises from a basis of interoperability, where C and C' accept the queries of D and give answers in a format understood by system D. Thus the relationship between the interacting systems is, barring further assumptions, asymmetric.

We posit that a query-based approach offers an elegant way around the difficulties of a translation-based approach to interchangeability and interoperability because:

(i) Queries, by D, restrict information transfer to only those data that are needed by the application of interest.
(ii) Queries let the creating system, C, determine the appropriate query result based on proprietary algorithms used to disambiguate idiosyncratic model information in M.
(iii) The query-based approach provides a rigorous foundation for developing broad communication and integration standards, in essence by providing operational semantics for fundamental, geometry-based activities.

We examine a core set of queries that are appropriate to support geometric modeling tasks and applications. The required information may or may not be explicit in the model file, so one may have to make additional assumptions and write code to reveal that information. This situation suggests a theme that we articulate as follows. If the model contains specific information, it should be possible to reveal this information and make it explicit. We call this activity *shape evaluation*. However, if the information is not present in the representation of the shape model, and if it must be derived and computed under additional assumptions or imputations, then we will speak of *shape comprehension*. We will point to this theme periodically.

1.2. Previous research on interoperability

Almost all earlier research on geometric interoperability can be characterized as data-centric by virtue of being focused either on format or specific representation conversions. A geometric representation can be thought of as a composition of geometric primitives by rules specific to a given representation scheme. In data translation, such a representation is transferred explicitly by various translators. However, in practice, the meaning of any representation is determined by the corresponding evaluation algorithms that usually also differ from system to system. Thus, conceptually, every geometric translation procedure involves three ingredients: primitive mapping, rule mapping, and possibly modified evaluation algorithms. While many of the primitives have been standardized in widely accepted STEP [2] and IGES [3] standards, representations in individual CAD systems remain incompatible. System-to-system translators are available in many cases, but they do not solve the fundamental bottleneck of interoperability. Perhaps the most widespread difficulty arises from the mismatch between the accuracy of geometric representation and the precision of the evaluation algorithms used in modeling systems. Attempts to deal with this issue include use of exact computation [4–6], modeling imprecision of data [7], methods for tolerant computing [8–10], and a number of heuristic techniques to "heal" the translated models [11–15].

A fundamental unresolved issue is that all data translation methods implicitly or explicitly rely on theoretical foundations laid out thirty years ago [16,17], assuming that sets of points and functions may be represented exactly by data structures and algorithms. These assumptions fail in the presence of numerical errors or approximations, as shown by researchers who proposed to extend the basic theory of solid modeling to account for geometric errors and tolerances [18,19]. In an effort to bypass the numerical issues altogether, a number of researchers proposed to approach interoperability problems in terms of higher level parametric feature-based representations that are largely symbolic structures with minimal numerical data [20–26]. Great progress has been made, but as of this writing, acceptable formal models are still lacking in a number of important areas, including blending, persistent referencing, constraints, and validity, to name a few. It was also observed that most geometric representations and algorithms may be recast in a canonical form using cellular representations [27–29]. In particular, researchers in [29] advocated a representation-neutral DJINN API based on cellular decompositions as an interoperability solution. While the approach is intellectually appealing, it is nonetheless impractical because it requires that a superset of all useful geometric operations is represented and exchanged in the canonical cellular format, by all interoperable systems.

Meanwhile, Shapiro showed [28,30] that, in the presence of a proper formal model, all exact representation conversions can be reduced to a small number of computations that included the construction of primitives, intersections, sorting, and point membership tests. This approach has been used to solve a number of challenging representation conversion problems, including boundary to CSG conversion [31,32] and maintenance of parametric families [24,26]. He also showed that the same generate-and-test paradigm applies in the presence of approximations and tolerances, provided that robust point membership tests can be performed against a formally defined standard. For example, this approach was effectively used to construct approximations of general sweep and unsweep operations based on a formally defined trajectory intersection test [33]. Independently, Hoffmann demonstrated that expensive and error-prone conversion of boundary representations models can be bypassed altogether, if such models may be tested against formally defined high-level parametric representations [34].

These and other recent results suggest that an effective approach to all "representation conversion" problems is not to convert them, but to compute on them via tests (or queries). The approach still requires a proper formal semantics, but this semantics is interpreted procedurally via computable queries. These observations were summarized in a recent report [1].

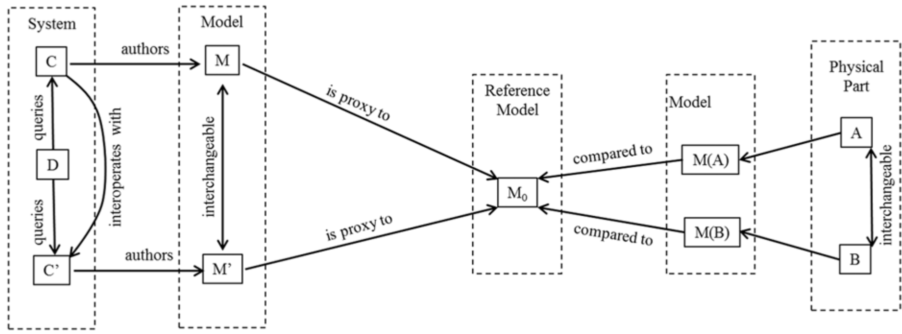


Fig. 1. Summary of the concepts.

1.3. Origins of interchangeability

Part interchangeability, in assemblies, is the basis for mass production and the economies of scale [35]. Originally, a set of gauges was used to determine whether a given part was fit to function in an assembly. A gauge is a simplest example of a device that performs evaluation queries on a given physical part. Later, the mathematical basis for part interchangeability came into existence with the development and standardization of geometric dimensioning and tolerancing (GD&T). This basis allows a precise specification of nominal shape and tolerance zones. Any part whose net manufactured shape is within the specified tolerance zones is fit to function and is interchangeable with any other such shape, assuming a correct nominal shape design.

We say that a part A, as a physical artifact, is *interchangeable* with another part B, also understood as a physical artifact, if A can be replaced by B without any compromise of form, fit, and function. Logically this relation is an equivalence. Let $M(A)$ and $M(B)$ be the point sets that model A and B exactly, with all imperfections. Since $M(A)$ and $M(B)$ cannot be congruent point sets, one may want to define interchangeability by positing that $M(A)$ and $M(B)$ are within some allowable, specified tolerance. But this definition is problematic because it is *not* an equivalence. Historically, the problem was solved by the notion of a "golden" master part, which is used to establish the equivalence between the interchangeable physical parts A and B. That is, *the parts are always compared with respect to an external common reference model M_0, and not directly to each other.*

In the early part of the 20th century, engineering drawings with dimensioning and tolerancing notations slowly replaced the need for keeping a physical master artifact, eventually leading to modern GD&T descriptions of all such reference models. In order to determine if a physical part A passes GD&T specifications, its geometric model $M(A)$ is obtained via measurement (*evaluation*) queries and then compared against tolerance zones via fitting (*comprehension*) queries. We make two crucial observations:

(i) The above traditional concept of parts interchangeability is responsive to the concepts of form, fit and function in an assembly. The concept is thus application-dependent; here on the application of assemblies.
(ii) The notion of interchangeability requires a common reference model that is determined by the application. In the case of GD&T the reference model comprises a set of tolerance zones and procedures for establishing them. Our query scenario is well-suited to this conceptualization.

Since an electronic CAD model M can serve a multiplicity of application domains, a generalization of *interchangeability* to models has to be relativized to the application domain. Moreover, the representation M, i.e., the CAD model, is (customarily) imperfect. In our case we stop at the proxy concept that we explain later.

The remainder is structured as follows. Section 2 defines interoperability, interchangeability and the proxy concept, and explains the concept of integration we consider. Section 3 presents the query hierarchy and illustrates evaluation vs. comprehension. Section 4 discusses several application scenarios, and Section 5 summarizes and concludes our paper.

2. Definitions and preliminaries

The diagram of Fig. 1 summarizes the concepts and definitions introduced below.

2.1. Reference model and proxy

Similarly to the time-tested approach to interchangeability in mechanical assemblies, a key concept of our framework is the reference model M_0 without which model interchangeability is not an equivalence: if model M' is within ϵ of model M and model M'' is within ϵ of model M', M'' need not be within ϵ of M. By contrast, all models that are within ϵ of reference model M_0 may be deemed equivalent. So the notion of equivalence rests on the role of the reference model M_0.

When a system C authoring a model M implements queries, consistent answers can only be to within a system-dependent accuracy ϵ_C. In many cases, however, M does not accurately describe the intended point set M_0. Defining the query semantics [36] that guides the interpretation of M and the association of M_0, is thus at best consistent to within ϵ_C. Hence limited accuracy is inherent in the reference model concept, which motivates the notion of a *proxy*.

Definition (Proxy). A model M authored in system C is a *proxy* for a reference model M_0 if M discloses the same semantic information up to accuracy ϵ_C when queried by C.

These assumptions put the onus of achieving a correct and consistent query implementation on the authoring system, where it belongs. Since different systems achieve different accuracies, queries include a parameter that requests an accuracy specified by the user. This user-stipulated accuracy ϵ can be at most ϵ_C. In the case of GD&T, for instance, ϵ is usually an order of magnitude greater than ϵ_C. In solid modeling applications, a system C may be a NURBS modeler capable of correctly answering point membership classification (PMC) queries to an accuracy of $\epsilon_C = 10^{-9}$ and C' may be a polyhedral modeler that can correctly answer PMC queries to an accuracy of $\epsilon_{C'} = 10^{-6}$. Both modelers understand PMC queries that are defined with respect to an idealized reference solid model M_0, and answer with a user-specified query tolerance ϵ_Q. In other applications, M_0 may be a scalar field and its level set may be queried with accuracy up to ϵ that is in turn limited by twice the sampling density.

2.2. Interoperability and interchangeability

Interoperability applies to systems and means that one system can work with the models of another system within an application domain:

Definition (Interoperability). Let C and C' be two systems and consider two models, M authored in C, and M' authored in system C'. We say that *C can interoperate with C'* if C can interpret and use correctly the queries that model M' supports.

Once again, correctness is understood with respect to the semantics of the queries associated with the application-specific reference model M_0 for which M' can serve as proxy. The fact that C can interpret correctly M' does not mean that C' can interpret correctly M. That is, interoperability is not a symmetric relation. Finally, the concept is restricted to a domain of queries understood by C' and by C. In the above example of a NURBS modeler C and a polyhedral modeler C', C can interoperate with C' but not vice versa.

Using interoperability of systems, it may be tempting (and sometimes practically necessary) to extend the notion of a proxy to computer models created in different systems. After all, if the system C can interoperate with C', then a model M' can be substituted for M, even if M and M' have been authored in different CAD systems. In the previous example a polyhedral model M' can be used as proxy for a NURBS model M provided $\epsilon_Q \geq 10^{-6}$. When using this approach, a model M created in system C must serve as the "master" reference model M_0 for all systems and applications that need to interoperate and exchange models with C. While this approach is often used in practice, it is limited by the fact that, by design, model M may not be mathematically sound, and the proxy relationship is asymmetric. In the query-centric approach, the systems query each other, each system disclosing data of the model it has authored, as requested. The concept of model proxy rests on the authoring system accepting and responding to queries in a way understood by both systems. The understanding must be based on a well-defined semantics ensuring correct interpretation.

The notions of model proxy and system interoperability can be combined to define the notion of interchangeable models as an equivalence relation that extends and generalizes the notion of interchangeable parts in assemblies discussed in Section 1.3.

Definition (Interchangeability). Let M and M' be two electronic models, authored in two different systems C and C' respectively. If M and M' are proxies for M_0, and C and C' interoperate, we call M and M' to be *interchangeable*.

Interchangeability of electronic models must be understood in an application context. The NURBS model M and polyhedral model M' may be interchangeable with respect to PMC queries with $\epsilon_Q > 10^{-6}$. But consider a 3D medical image M_0 sampling a scalar field with a grid spacing of δ, and two models M and M' of a level set of the field, supporting PMC queries with an accuracy of ϵ. Assuming $\epsilon > 2\delta$, [37], both M and M' can serve as proxy for M_0 and are therefore interchangeable. As another example, two proxy models M and M' of a reference model M_0 may be deemed interchangeable if their structural performance is sufficiently close to that of M_0. In this case, depending on the problem and analysis method, specific shape features and model representation may be considered irrelevant. Thus, finite element mesh models, simplified boundary representations, or a detailed solid model sampled at some coarse resolution may all be considered interchangeable between systems that interoperate via queries.

Last but not least, when systems import and export models in the neutral STEP format, there is also the need for a well-defined semantics of M_0. Here, too, we would define proxy models analogously to the query-centric definition above. We note that it, too, is a matter of system capabilities. We thus understand the various conceivable forms of proxy models as differentiated by the degree of system interaction and data granularity. In the data-centric approach the query is simply "give me the model, say in STEP format". Once the model has been delivered, the job of the authoring system is done. This is a coarse-grained data interchange. In the query-based approach, there is an incremental information exchange that is finely-grained. Restricting the semantics to a set of queries allows a more nuanced semantics than for the data-centric approach.

2.3. Integration

So far we have mainly considered model interchangeability, an equivalence where the application context narrows the semantic requirements. We now explore model complementarity.

Definition (Integration). Let C and D be two heterogeneous systems, with C a CAD system. Then *C and D can be integrated into system S if S can interoperate with both C and D in order to acquire new functionality.*

Note that C or D can play the role of S, in which case we talk about integrating D into C or vice versa. System integration deals with combining systems to accomplish a task beyond the individual system's capabilities. For instance, we might integrate a CAD system C with MatLab. MatLab has no notion of CAD models, but by querying C it can determine the mass properties or compute the convex hull of CAD models in C.

Similarly, consider a CAE system D that performs engineering analysis, such as linear static analysis, on a geometric model M authored in a system C. The data-centric approach to integration requires converting M into a format acceptable to D, such as a finite element mesh. We will explain in Section 4 how, in a query-based approach, integration is achieved by replacing all geometric computations within D by queries to system C. Of course, this integration requires that C can interoperate with D, with respect to a common reference semantic model, in this example a valid solid.

2.4. Models: incidence structures with attributes

In this paper, we will assume that a model M is a collection of shapes along with incidences and attributes. For instance, in a boundary representation the shapes are faces, edges and vertices that, based on the incidence and adjacency structures, imperfectly describe and interpret an ideal point set M_0 that is the boundary of a 3-dimensional semi-analytic set, the *mathematical model* of Requicha [16].

The information content of a CAD model M has to support many different *views* (information structures) of the artifacts to be manufactured, maintained and eventually disposed of. At the various stages in the lifecycle, different shape information is required. Beyond incidences and adjacencies, the shapes in M are associated with attributes and relationships. Common examples include: mechanical assemblies, composite material structures, and CAD boundary representations with GD&T information. Instead of thinking of M as subdividing and relating parts of M_0, we think of M as a collection of shapes.

If different incidence structures refer to the same point set M_0, we do not require that the incidence structures are compatible. For instance, if M and M' refer to the same shape M_0, then a face of M need not be contained in a face of M' or vice versa. Instead, we may think of a collection of different shape models M, each potentially supporting different stages and requirements in the life cycle.

Throughout, we restrict to queries pertaining to shape and do not delve deeply into the application or process information that takes its cue from shape. However, we note that geometric incidence structures also carry large amounts of application-specific knowledge, usually in a form of attributes associated with the shapes of M. Common examples include GD&T information associated with boundary representations, constraints associated with mating conditions in assemblies, and material and manufacturing process information associated with composite layers. In order to support such applications, an incidence structure would assign to the shapes of M identifying tags and would associate with some of these so-identified shapes labels that link to additional information and relationships pertaining to the application.

We note that the same metaphor applies to other application domains, e.g. medical models, building information management, etc. Moreover, the model M itself may be referred to by a tag that names M in its entirety. Thus, when passing a shape query to a model M, an argument M can be understood more generally to be an associated tag.

3. Query hierarchy

We present a query hierarchy to demonstrate how to solve the interoperability, interchangeability, and integration problems. We will make a number of specific assumptions that are reasonable in the context of design and manufacturing applications. Note, however, that many of these assumptions may be modified if needed. For example, we will assume that all representations deal with subsets of Euclidean space $\mathbf{R}^d$, where typically $d = 2$ or 3. The queries and the hierarchy are based on well known results in geometric and solid modeling, which we cite as appropriate. The hierarchy is organized into levels, by increasing degree of comprehension: the higher level queries assume the existence of lower level queries and in fact may be implemented in terms of them, with additional assumptions. Our descriptions below aim for clarity instead of stodgy detail.

3.1. Assumptions

We assume throughout that the model M of the queried system C is a logical incidence structure over a collection of one or more shapes M_i. Without loss of generality, lower levels of queries will assume that M is a single shape, while the highest level in the hierarchy assumes that M is an incidence structure involving any number of shapes with attributes, as explained in Section 2.4. For models M of an incidence structure, we will assume that M can refer unambiguously to its constituent models M_i by tags that can be queried. The constituent models may also contain attributes with labels that are relevant to applications. Such labels might refer to surface finish, material properties, and GD&T, to name a few. At the lower levels of the query hierarchy, however, it suffices to assume that M refers simply to a reference point set M_0.

In CAD, a (single) shape M represents a set M_0 that is a connected compact k-manifold with boundary, that is tamely[1] embedded in an ambient space. Unless stated otherwise, this is the Euclidean space $\mathbf{R}^3$. Generally speaking, we assume that the metric properties are determined by the properties of the reference space. The metric properties of M need not be the same as the metric properties of the ambient space, for example, when M is a curve or surface embedded in $\mathbf{R}^3$.

Now M is a possibly imperfect computer representation of M_0, the reference model. Assume that the authoring system C of M can correctly answer queries of M. When the queries are numerical, C can answer them to a specific resolution ϵ_C.

[1] Tame embeddings ensure that the represented shapes are physically realizable [16,38]. In manufacturing, shapes are also assumed to be semi-analytic sets with piecewise smooth boundaries [16].

Fig. 2. The handle of the Berkeley Pitcher, modeled as a surface, requires determining δ_i using geodesic distance, whereas δ_e is determined from the Euclidean distance. Parts of the MATs are as shown, of the surface and of the complement Euclidean space.

For example, a point membership query that returns $q \notin M$ means that $q \notin M_0$ and the distance of any boundary point q_0 of M_0 from q is at least ϵ_C. If a nearest point q_0 on the boundary of M_0 has distance less than ϵ_C, then the query returns *on* the boundary of M. The query may specify a coarser resolution ϵ_Q. An incorrect model M either does not have an interpretation of M_0, or point-membership queries return inconsistent answers. In an abuse of language we say that M supports a query Q to mean that the authoring system C of M supports Q.

3.2. Level 0: dimension and bounds

The dimension k of M is its intrinsic property. By definition, $k \leq d$, the dimension of the ambient space. We assumed so far that M must be dimensionally homogeneous, which means, in $\mathbf{R}^3$, M can be only a single point, connected curve, surface, or solid. Disconnected sets and heterogeneous structures are represented by logical incidences between basic shapes.

Dimension Query DQ(M): returns the dimensions k of M and the dimension d of the ambient space $\mathbf{R}^d$.

In most cases, certain bounds on M are needed that guarantee correctness of a computation involving local queries. They are a bound on the extent of M, such as a bounding box, and a separation distance δ_0 that informally states that no two nonadjacent points of M_0 are closer than δ_0. This notion may be made precise by defining $\delta_0 = 2\min(\delta_i, \delta_e)$, where δ_i and δ_e are the minimum feature sizes of M's interior and exterior respectively. The minimum feature size of a set X is in turn defined as the minimum distance between ∂X and its medial axis [39]. Note that the medial axis of X is computed using its metric properties; thus the medial axis of a surface M embedded in R^3 will be computed in terms of geodesic distances, while the medial axis of its complement will be computed using Euclidean distance (Fig. 2). If M can deliver consistent answers about M_0 up to the resolution limit ϵ_C, it follows that $\delta_0 > \epsilon_C$. Moreover, the native coordinate system underlying M must be understood so the queries can be meaningfully formulated. We accomplish both with the bounds query.

Bounds Query BQ(M): returns the resolution ϵ_C, the separation distance δ_0, and a bounding volume X, of M_0. X is an axis-parallel box, specified by two vertices delimiting a main diagonal of the box and given in Cartesian coordinates in the native coordinate system of M. C is the authoring system of M.

3.3. Level 1: distances

We assign to each point p that is not in M_0 the shortest Euclidean distance to a point $q \in M_0$. If p is in M_0, then its distance is less than the resolution limit ϵ_C. If p is not in M_0, then a nearest point $q \in M_0$ is on the boundary ∂M_0.

Distance Field Query DF(M, p, ϵ): Given point p and distance ϵ, return a nearest point of M provided no point in the ϵ neighborhood of p has more than one nearest point on M.

If we are near singularities, i.e., near the medial structure of the complement space, there can be several equi-distant points on M. Spheres and surface elements of revolution have an infinite number of nearest points to points in the center or on the axis of revolution. If the query point is near such singularities, the query returns a nearest point q plus an indication that there may be more than one nearest point. As mentioned, in GD&T ϵ is typically an order of magnitude larger than ϵ_C.

It is a basic fact that the distance field exists for any closed set, and conversely, a set of points satisfying $f(p) \geq 0$ for any real valued function f is a closed set [40]. This means that even at this very low level, we already have a model of informational completeness: a distance field defines a set M_0 with precision ϵ_C, which can be reconstructed within the bounding volume and feature details that are larger than δ_0. In fact, distance fields are used directly to represent shape and physics information in geometric and level set methods [41,42]. It is possible to (re)construct a model in this way, but higher level queries allow more efficient algorithms to do so.

3.4. Level 2: point sampling

We assume that M_0 is a point set on which d-neighborhoods are defined by distance. If p is a point in the space in which M_0 is embedded, then the ϵ-neighborhood of p is simply the set of all points q at distance less than ϵ from p. The *point-membership classification query* (PMC) is defined as follows:

(1) **Point Membership Classification Query PMC** (M, p, ϵ)

The query is defined for $\epsilon \geq \epsilon_C$. It returns *in* if the ϵ-neighborhood of p contains only points in M_0 and returns *out* if no point in the ϵ-neighborhood of p is contained in M_0. The query returns *on* otherwise. In this latter case, a classical version of point membership test also returns a representation of the neighborhood of p with respect to set M_0, usually in terms of normal directions at the boundary. In an effort to keep queries as simple as possible and to avoid complex representations, we will instead recover the neighborhood information through combinatorial and differential queries at Level 2.

An iterated PMC query of the first type can construct an ϵ-cover[2] of M_0 if we have a bounding box of M_0. An algorithm for constructing such a cover could proceed by exhaustive search within the bounding volume. Since ϵ is an argument to the query, more efficient algorithms can be devised that adaptively search for points near the boundary of M_0. In particular, an iteration can determine the distance of a query point p from the boundary of M_0. This discovery process may be of limited efficiency, however, owing to a potentially large volume of queries to be issued. Indeed, in many representation formats (for example, in all parametric representations of curves and surfaces) M can more efficiently construct and return an ϵ-cover of M_0. Thus we also include a generative version with the second query type.

(2) **COVER** (M, ϵ) The query returns an ϵ-cover of M.

When $\epsilon < \epsilon_C$, both forms of the PMC query return *unknown*. With point membership and ϵ-cover queries, we can claim a slightly higher level of informational completeness. If the extent of M_0 is known, then it can be approximated by a point cloud with $\epsilon \geq \epsilon_C$ accuracy using PMC queries. Such ϵ-cover can be used directly by many applications or may be used to construct either combinatorial or differential models of M_0. The reconstruction is essentially a comprehension procedure that assumes additional properties of M_0. These properties may be established by local geometry queries next.

3.5. Level 3: local geometry

At Level 1, we only worked with ϵ-neighborhoods. We could classify points and find their distance from each other, but without additional information we did not know whether two nearby points were adjacent in a simplicial approximation of M_0. We now assume that M has basic local geometric and topological properties. In the discrete case, we consider a local simplicial approximation of M_0. In the continuous case, we consider approximate differential properties. In either case, M_0 admits these properties and M supports querying them.

3.5.1. Level 3A, discrete geometry

The following queries assume that M can answer queries about the local topology correctly but only to a particular resolution ϵ_C. As before, this bound necessarily implies that the smallest distance δ_0 between two points of M_0 that are not adjacent is not smaller than ϵ_C.

Combinatorial Neighborhood (CN) queries.

(1) **CN**(M, p, ϵ, k):
Given a point p in or on M_0, return the combinatorial neighborhood of size ϵ and dimension k, of a local simplicial approximation of the ϵ neighborhood of p.

(2) **CN** (M, K, ϵ, k):
Given an ϵ cover K of M_0, return a simplicial approximation of dimension k of M_0.

The CN-(1) query can be used to discover the local topology of p. The second query, CN-(2), is a generative query that returns a simplicial approximation. It is motivated by efficiency considerations, but still conveys only local topological information. It may seem that the information content of the Level 2 ϵ-cover query and CN-(2) is similar, but in fact the CN queries are at a higher level because they require additional assumptions. For example, an ϵ-cover may be used to approximate the (manifold) topology of M_0 up to homotopy using a Cech complex based on intersection of the ϵ-neighborhoods, or by a Rips complex based on the distances between the adjacent points [43]. The resulting complexes have different mathematical and computational properties, but both are useful in practical applications.

3.5.2. Level 3B, differential geometry

Here we assume that M can determine differential properties such as normals and curvature. When M contains only a piecewise simplicial approximation, then such differential properties could be supported approximately, derived from the local neighborhood.

Differential neighborhood (DN) queries.

(1) **DN**(M, p, k, r):
Given a point p in or on M_0, return the differential information through order r for the k-dimensional submanifold of M.

(2) **DN** (M, K, k, r):
Given an ϵ cover K of M_0, return for each point in the cover the differential information through order r for the k-dimensional submanifold of M.

These queries assume that M_0 is a differentiable C^r-manifold and its differential structure may be recovered at any point $p \in M_0$.

[2] A set of sampled points $\{p_i\}$ is an epsilon cover of set X, if every point $x \in X$ lies within ϵ-distance from some sample point p_i.

Ideally, differential information should be recovered in coordinate- and representation-independent form. For example, if $k = 1, r = 3$, the query would return tangent, curvature, and torsion for a point on a curve; when $k = 2, r = 2$, the query would return tangent (or normal) and principal curvature information. When such intrinsic information is not available, it is reasonable to expect that DN queries return differential information of the manifold's embedding in the ambient space in a form of a Taylor expansion at point p. In this case, a query such as $\mathrm{DN}(M, p, 2, 3)$ would return a list of coefficients of derivatives through order 3 in the Taylor series expansion of the embedding defining surface M. Intrinsic quantities, such as surface normal and curvatures, may be easily recovered using their standard definitions.

The DN queries parallel the CN queries, and the two should yield the same information in the limit as $\epsilon \to 0$, allowing us to recover not only topological but also differential structure of M_0. Normals, curvature, and other differential information may not exist for singular points. In some cases, differential information may be replaced by equivalent generalizations, such as generalized normals or local cone approximations when they are defined and exist.

3.6. Level 4: global geometry via intersection queries

Many common tasks in geometric modeling and applications require knowing global properties of M_0. Examples include collision and containment detection, ray casting, and surface–surface intersection, to name a few. Global properties can be computed, in principle, from lower-level local queries, at least approximately and with additional assumptions about M_0. Thus, for reasons of convenience and efficiency it is helpful to define higher-level queries. In a data-centric approach, global model information is obtained from the model M given in a particular CAD-model format. Instead, we want queries and their answers to be put in the simplest, most basic representation, and with a minimum of algorithmic requirements. This, in turn, requires identifying queries that acquire the global shape information in a variety of situations.

We propose that this role is played by intersection queries. Formally, this argument is based on the observation that most global computations (boundary evaluation, collision detection, ray casting, etc.) tend to reduce to a finite number of Set Membership Classification procedures [44] formulated as follows. The candidate set X is being classified against the reference set S into three sets *XinS*, *XonS* and *XoutS*. Each of the three sets is computed by regularized intersection of X with the reference set S, its boundary, or complement, respectively. The results are regularized in the topology of the candidate set X. As explained in [44], Set Membership Classification is a natural generalization of the point membership test. We adapt set membership classification as a basic query by treating M as a representation for the reference set M_0, and choosing a simplest possible generic representation for the set X: an r-dimensional simplex.

Intersection Classification IC(M, P, ϵ, r): Given points $P_r = \{p_0, \ldots, p_r\}$ that form a r-simplex P, the query returns a collection of r-simplices contained in P that, together, approximate $M \cap P$ with ϵ resolution, usually measured in terms of the Hausdorff distance.

The query clearly subsumes the line/solid classification $\mathrm{IC}(M, \{p_0, p_1\}, \epsilon, 1)$ in which the segment (p_0, p_1) is partitioned into segments according to how the segment intersects M_0. In practice, we may not need to restrict to simplices and simplicial complexes. For instance, it may be convenient to include a box/solid classification even though it can be implemented by 5 or 6 simplex/solid classification queries. Ideally, the queried system answers the supported queries in ways that are best suited to its, possibly proprietary, internal logic.

3.7. Level 5: logical shape information

So far, our queries conceptualize M_0 just as a point set approximately represented by the model M. What structure is queried at those levels is primarily driven by submitting a point set of interest and obtaining how it relates to M_0 in a basic way: e.g. is a query point in, on or out of M_0? If it is in, what is its neighborhood; if it is out, how far away is it; etc. Queries with higher-dimensional volumes concern analogous questions, for instance how the query volume intersects M_0.

Now we turn to a more differentiated view that acknowledges that a model M can have a logical structure and that this structure, for instance, decomposes M_0 and relates the constituent parts by incidence information in M. The parts of M_0, organized by M, have identifying tags which we denote as *ID*s. We may query what ID a particular point has, but we can also query what incidences there are between the IDs in M.

Identification Query IDQ(M, p, ϵ): Given a point p that is in or on M_0 with resolution ϵ, return the set of IDs of the parts containing p and the dimensions of these parts.

For example, if M is a boundary representation and p is on the boundary of M_0, the label returned could be the name of the face or edge containing p, or of the vertex p. Because we assumed that M_0 is a decomposition (and not necessarily a partition) into constituent closed sets (parts), p may be contained in more than one part. When multiple IDs apply to p, the list of IDs of all containing entities is returned. Once such an ID of a part is known, we can request incidence information:

Incidence Query INC(M, J, k): Given the ID J of a k-dimensional part of M_0, return the IDs of all k-dimensional parts incident to J in M.

These two queries are sufficient to reconstruct incidence information in any incidence structure, including commonly used simplicial and cell complexes, assembly and constraint graphs and tolerance tree models, heterogeneous material structures, finite element meshes, as well as design/manufacturing feature interaction graphs, among others.

3.8. Level 6: knowledge aggregation

Recall that we have limited the space of represented incident structures to geometric information. Net shape and incidence structure, such as may be represented by M, is one component of the information structures that are employed to represent products. The shape and incidence model is augmented by other information, pertaining to features, surface finish, typical loads during product operation, etc. Much of this information can be conceptualized as attribute annotations associated with specific shape elements. Similarly, incidence relations may refer to datums, mating or interference surface features in an assembly, or indicate a critical distance constraint in a mechanism. Layering in composite materials, adjacency in open cell complexes, image segmentation, or datum relationships in GD&T specifications are other examples of many possible incidence relationships. That is, the application context modulates the needs and semantics of the comprehension-richer queries.

The syntax and semantics of the attribute information are determined by an application and should be consistent with accepted standards [45]. Examples of useful standardized attributes include type and degree of surfaces, material properties, GD&T specifications, and other annotations. We will see examples of how attribute information may be used in Section 4. But since our main focus is on geometric queries and interoperability, we will not attempt to deal with the semantics of the attributes.

4. Example applications

We demonstrate how the query-based approach may be used to implement a number of common tasks in computer-aided design, manufacturing, and simulation. Our goal is not to propose optimal solutions or to improve on existing algorithms, but to demonstrate the broad applicability of the query-based approach to solving problems of interoperability, interchangeability and integration.

4.1. Approximate shape interoperability

Consider first the classical problem of constructing an approximate polyhedral model of a homogeneous 3D solid. The model M to be so approximated has been authored in a system C that answers the queries discussed before, and we assume that C can answer with an accuracy of ϵ_C. The task of (re)constructing M' in another system C' corresponds to the simplest example of interoperability—that of net shape data acquisition. This may be useful in a variety of practical situations, for example, when a reconstructed model M' can be prototyped using layered manufacturing equipment without revealing any information about the design process or intent. Another application relating to long-term archival is discussed in Section 5.2. In all of these applications, M' serves as a proxy for the nominal solid model M_0 represented by M. A simple algorithm could proceed as follows:

Algorithm 4.1

(i) Establish a bounding volume V for M' using a bounds query; **BQ**, Level 0. Note that we assume that M' is three-dimensional, but its dimension can be established using a dimension query; **DQ**, also Level 0.
(ii) Sample V with a regular grid of density $\epsilon > \epsilon_C$, establishing for every grid point whether it is in, on or out of the solid represented by M, using the point-membership classification queries, **PMC**, Level 2.
(iii) Assemble the polyhedral boundary of the approximation for model M' using the usual marching cubes method [46] or another polygonization algorithm.

In practice, Steps 2 and 3 are best combined, using the marching cubes paradigm to restrict sampling to the grid cells intersecting the boundary of M. Seed points at which to begin assembling each connected component of the boundary can be generated using distance field queries. As pointed out already, it is more efficient to let system C deliver an ϵ-cover (**COVER**, Level 2), if C supports this query for the boundary of M, obviating Steps 2 and 3 of the algorithm.

What accuracy can be achieved? According to the Nyquist–Shannon Theorem [37], a sampling density of at least $\delta/2$ is required to reconstruct a signal whose highest frequency is δ. Given that C can deliver an accuracy of ϵ_C, the smallest distinguishable feature size is therefore $2\epsilon_C$. Note, moreover, that sampling can only approximate sharp edges and conical singularities, as they have infinite frequency. However, when the ϵ-cover is constructed by the authoring system C, then those features can be approximated as sharp polyhedral edges and conical singularities.

The approximation of M', whether done with PMC queries constructing an ϵ-cover or by obtaining a cover by query from C, can be generated for any shape model of any dimension. For instance, manifolds for tool path design can be so approximated. Thus already low-level queries are flexible and quite useful. Recall from Section 3.2 that feature size is defined in terms of bounds on distances between the boundary of M and the medial axes of its interior and exterior. In fact, we could even approximately reconstruct the medial axes using a combination of point membership and distance queries to some accuracy ϵ_Q and determine the smallest feature size. Of course, the correctness of the medial axis reconstruction itself is subject to the same sampling limitation, albeit at higher frequency. These limitations are reasonable manifestations of the fact that low-level queries make very few assumptions about represented properties of M. In contrast, below we show that higher level queries may be used to reconstruct a model M' more accurately.

4.2. Precise model interoperability

We consider whether and how some of the higher-level queries permit highly accurate reconstructions of model M under the assumption that the authoring system C can query M with precision ϵ_C. If $\epsilon_C = 0$, the reconstruction procedure would be exact. We illustrate this application with an example.

A trivariate polynomial of degree m has $c(m)$ coefficients, where $c(m)$ is $O(m^3)$ [38]. Thus, an algebraic surface of degree m is fully determined by $c(m) - 1$ surface points, since we may scale the surface equation by a constant. An attribute query (Level 6) can ask the degree of an algebraic surface, followed by a sequence of queries to generate the required number of points. The specific sequence of queries may depend on queries supported by the authoring system C'. For example, if surface M_i is a parametric surface, an ϵ-cover query (**COVER**, Level 2) will generate a sufficient number of candidate points; if the ϵ-cover is not supported, the points on the surface could be located by a binary search using point membership and distance queries; or if C' supports line/surface intersection queries, (**IC**, Level 4), then randomized line/solid queries will determine $c(m) - 1$ surface points with high probability. In all cases, once the required number of sample points is obtained the surface may be reconstructed with high accuracy using the representation in the receiving system.[3]

This process may be combined with incidence queries (**INC**, Level 5) to support what is essentially a lossless transfer and reconstruction of any incidence structure M representing piecewise-algebraic model M_0, including a boundary representation. It should be clear that the incidence queries are sufficient to determine dimension and adjacency information for all faces, edges, and vertices in any boundary representation. The geometric carrier of each boundary cell (edge or face) can be reconstructed precisely following the above procedure. When the type and degree of a cell may not be available, it is reasonable to expect that its geometry can be deduced from neighboring cells: intersection of incident faces for an edge, and interpolation of incident edges for a face. Additional information on accuracy of intersection or type of interpolation may be available as an attribute associated with the cell. In all cases, in contrast to data-centric interoperability solutions, the query-based approach does not require standardized representations or file formats beyond the minimal attribute information.

4.3. Query-based interchangeability with GD&T

The examples discussed above demonstrate that an information exchange between systems can provide a high level of interoperability. However, no universal claims of model interchangeability can be made based on such arguments alone, because the notions of interchangeability and proxy require semantic models with respect to which these notions can be defined rigorously and validated. For example, a proxy relationship may require that the Hausdorff distance between a computer-represented model M and the reference M_0 is bounded by some value τ. This would also imply that all models that represent the

[3] Note that error estimates would depend on the distribution of the points in addition to m and ϵ_C.

same point set within the Hausdorff distance of M_0 are interchangeable (see also Section 5.2). Similarly, we could postulate that the precise reconstruction described above procedurally defines yet another type of interchangeability of piecewise-algebraic boundary representations. Other notions of interchangeability are useful in the context of specific applications. In this section, we consider how the query-based approach may be used to test for one of the more common and important notions of interchangeability in mechanical assembly as defined by GD&T standards [47].

The purpose of GD&T is to establish part interchangeability in mechanical assemblies. Historically part interchangeability was established first with gauges, measuring critical dimensions and functional features. Later, gauges were replaced by conceptual (and sometimes physical) master parts that embodied the specifications. With the development of rigorous mathematical foundations for dimensioning and tolerancing, these tools and notions were supplanted by a mathematical model $M_0(X)$ that specifies functional features, dimensions, and allowable variation of form for every part X in the assembly.

In modern CAD systems, M_0 is represented as an incidence structure M that is typically an augmented boundary representation; the GD&T information is stored textually as a set of attributes associated with faces. The task of (physical) metrology is to determine whether a given physical part X belongs to the equivalence class of interchangeable parts specified by M_0. We do not aim to improve on the current metrology procedures; rather, our goal here is to demonstrate that such procedures are readily formulated and implemented using queries.

Roughly speaking, the semantics of a given GD&T specification is determined by the face attributes in M, which designate certain faces of the solid to be datums and certain faces to be features via feature control frames. Each frame controls size, position, or form of the associated faces in terms of allowable tolerance zones that may be formally constructed from the nominal boundary representation following the rules specified by the standard [47]. Some of the tolerance zones, (for example, position and orientation tolerances) refer to other faces of M_0 that serve as datums; other frames, for example, flatness and other form tolerances, are constructed with reference to the nominal face itself. Multiple feature control frames may be associated with a face; each frame dictates conformance of the physical part to a set of tolerance zones and must be verified independently from other requirements. For additional details, the reader is referred to [47]; note, however, that the semantics of the GD&T standards (which determine what the reference model M_0 is) is continuously evolving to reflect improved manufacturing and metrology practices.

A typical GD&T requirement may be verified by a two-stage procedure. First, a physical part X is sampled, or *evaluated*, using a coordinate measuring machine (CMM) or another measuring device. The measurements are segmented into subsets, each subset sampling a particular feature, as stipulated by the feature control frame. This evaluation process typically results in a set of segmented point clouds. In the second step, these sample measurements are then used to fit shape features, such as datum planes, cylinders, spheres, or measure size of the implied tolerance zones. Thus a shape comprehension process, where higher-level shape semantics is derived from lower level queries, produces a model $M'(X)$ that needs to be compared to the specification M according to the formal semantics $M_0(X)$.

As a concrete example, consider how conformance to a flatness tolerance may be verified for a flat surface X on a physical part [48]. The specification model M may be queried for its nominal geometry, type (degree 1), and flatness tolerance specified as an attribute. The validation process requires obtaining a cloud of points measured on X that are filtered and registered with respect to M by repeated point membership and distance queries. Once the desired cloud of points is obtained it can be used as an ϵ-cover of the surface $M'(X)$, or to simulate $M'(X)$ via a least-square fitting procedure. It can also be used to measure the implied width of the tolerance zone containing M' and to compare it with the size of the tolerance zone specified in M.

The same conceptual two-stage process may be used to validate all other types of GD&T specifications, but the two stages are often used iteratively and adaptively, and the comprehension processes may require queries from any level in the query hierarchy. For example, when a tolerance specification for $M(X)$ refers to a datum feature Y, that datum feature must be evaluated and comprehended before feature X can be verified. For free-form surface features, non-uniform sampling, partitioning, and adaptive registration are usually required based on differential (curvature) and distance queries on M during the evaluation stage of the procedure [48].

Thus, for physical metrology, physical part interchangeability rests on measurements that are ultimately vetted by the semantics M_0 of GD&T specification M. When replacing a physical part X with a specific electronic CAD model $M(X)$ authored in some CAD system C, these measurements neatly correspond to queries.

4.4. CAD/analysis integration

Our final illustration of the query-based approach to interoperability is a fully implemented query-based solution of one of the most difficult integration problems: CAD/analysis integration. Analysis systems (FEA, CFD, etc.) and CAD systems represent geometry incompatibly [49]. The most common and industry-accepted data-centric approach to engineering analysis is to discretize the CAD geometry by a boundary-conforming mesh. This translation is computationally challenging and error-prone. The mesh must satisfy certain quality measure criteria which depend both on the type of analysis problem and also on purely geometric considerations, such as aspect ratio, element size, and so on. Fully automated generation of high quality meshes remains a research problem [50].

Instead of meshing, we adopt a different query-based principle that delegates all geometric queries to the native CAD model and all analysis computations to a separate analysis model. This principle underlies RFM [51], WEB-splines [52], immersed boundary [53], and many other meshfree methods [49] that are becoming increasingly important. The viability of this approach has been demonstrated recently by a fully automated interoperability solution for structural analysis [54]. The approach requires (1) formal semantics of the analysis problem suitable for queries; (2) identification and implementation of queries; and (3) a system architecture supporting a query-based approach to the analysis. We describe these components briefly; see [54] for details.

Semantics Fig. 3 illustrates the approach for stress analysis. The CAD model is a boundary representation and the analysis model is a linear combination of B-splines on a uniform non-conforming grid. This approach is a modified Finite Element Method (FEM) based on a generalization of Kantorovich's method [55,51]. Briefly, the solution of the displacement $\boldsymbol{u}$ has the general form

$$\boldsymbol{u} = \sum_{i=1}^{n} \boldsymbol{C}_i \boldsymbol{\eta}_i + \boldsymbol{u}^*; \qquad \boldsymbol{\eta}_i = (\omega^x, \omega^y, \omega^z)^T \chi_i, \tag{1}$$

where ω_1, ω_2 and ω_3 are smoothed distances to the boundary in the principal directions respectively and multiply B-splines χ_i that are located on a uniform grid in space. The function $\boldsymbol{u}^*$ interpolates the boundary conditions on Γ_u, and vector valued coefficients $\boldsymbol{C}_i$ of basis functions η_i are to be determined. Note that the function $\boldsymbol{u}^*$ is constructed directly from the boundary conditions, without meshing. This is done by direct interpolation based on inverse-distance weighting [56]. Secondly, the basis functions η_i are B-splines multiplied by distance functions, so they vanish on the

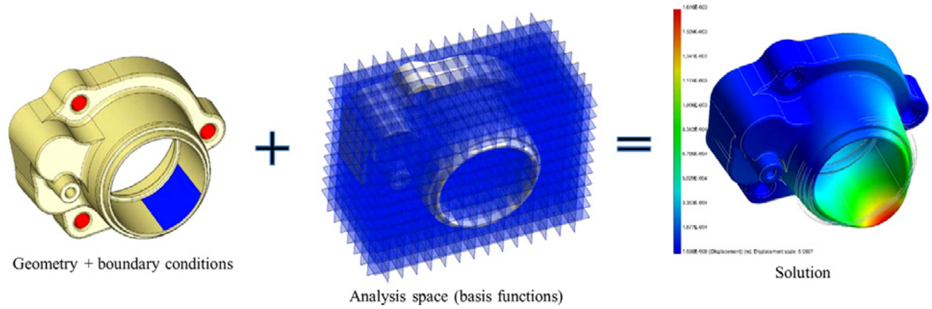


Fig. 3. CAD/analysis Integration Concept: all geometric computations are delegated to the CAD system and are integrated into the analysis model at run time.

boundary of the CAD model where Dirichlet boundary conditions (such as fixed restraints) are applied [52,57]. Substituting this expression for $\boldsymbol{u}$ into a weighted residual statement, we can derive a system of linear equations as shown in [54]. The solution is substituted into expression (1) and gives an approximate solution $\boldsymbol{u}(x)$ of the differential equation satisfying the specified boundary conditions. Complete details of this formulation and comparison with classical FEM can be found in [54]; here we simply note that the application semantics requires M_0 to be a valid solid with well defined point membership and distance-to-boundary functions.

Queries. The above formulation cleanly separates geometric and analysis computations. Required geometric queries are:

– Query boundary conditions (loads, restraints, etc.)—attributes associated with elements of the boundary representation (level 6 queries).
– Compute distances (**DF**, Level 1) to the restrained faces for multiplying B-spline and distance functions; i.e., distance-to-face computations for computing $\boldsymbol{u}^*$.
– Point membership test (**PMC**, Level 2) for sampling points in the B-spline support, for the volumetric integrals.
– Point generation (**COVER**, Level 2) for the surface integrals and visualization of the computed results.

Some of the above computations may be implemented more efficiently using more advanced queries, for example, Box/Solid intersection and Ray/Solid intersections (**IC**, Level 4). But fundamentally, all of such computations may be implemented in terms of point and distance computations [44,30].

System. The above analysis suggests a computation procedure that combines the two representations (geometry and analysis) at run time using appropriate interoperating queries in order to compute the solution of an analysis problem. The CAD model is never explicitly converted into a mesh or restrained by meshing. Implementation details can be found in [54]. A commercial-strength implementation of this procedure for structural analysis of popular Rhinoceros models has recently become available.[4]

For concreteness, consider how the system would perform specific geometric tasks and queries required to solve the analysis problem with approximately $O(10^5)$ basis functions. Roughly, $O(10^6)$ distance and possibly point-membership computations (PMC) would be used to sample each restrained face to generate a distance field. Integration algorithms rely on adaptive subdivision in order to generate the Gauss points at which the functions will be sampled [58]. Assuming $O(10^5)$ tri-variate B-splines, this can be achieved in several ways. An efficient method would require $O(10^5)$ Box/Boundary intersections and $O(10^5)$ Ray/Boundary intersections, but these queries can be also reduced to repeated PMC tests. Volumetric integration requires roughly $O(10^7)$ function and derivative samples, while surface visualization will need $O(10^4)$ boundary samples. The procedure reduces to massive sampling, ideal for massively parallel implementations, and solving a linear system of equations.

5. Conclusions

5.1. Summary and main points

The advantages of our query-based approach include overcoming some of the thorniest obstacles to the data-centric approach to CAD interoperability, namely expressing, in a neutral format, the underlying (often incompatible) assumptions, proprietary algorithms, and heuristics used by the authoring system to interpret imperfect native CAD model data.

Queries overcome these barriers by making the authoring system responsible for interpreting the original models about specific properties, such as point/solid classifications for which an external semantics can be defined. The specificity, and focus, of the queries ameliorates the semantic difficulties that both researchers and standardizing bodies have labored over for many decades. More than that, based on a sound semantic foundation, we can judge correctness and accuracy of the vendor's implementation, an aspect traditionally approached heuristically by the end user with the question "how many of my models, translated into STEP or another standard, are flawless?" and "how difficult is it to repair those models?" Other advantages include the ability to devise new queries for advances in manufacturing processes, re-implementation of queries as better algorithms are found and other platforms are introduced. *Queries are agile.*

Long-term archival of CAD models is a serious problem that is not directly improved by the query-based approach. As platforms change over time, and as CAD systems evolve, the interpretation of CAD models may change. Queries quarantine such seismic shifts and lay the responsibility squarely where it belongs: with the authoring system. Nevertheless, some more speculative applications of queries we suggest below might be helpful for long-term archival.

In Section 3 we have offered some specific queries, focusing on geometric data. The point of that exercise is to show possibilities some of which are sketched in Section 4. The queries were roughly organized by the metaphors of evaluation and comprehension, a rough measure of detail knowledge vs. global characteristics. As we mentioned in the context of metrology, both paradigms are in play all the time, although we have not found an explicit identification elsewhere. Other example applications in Section 4 illustrate the flexibility of the queries we described. More queries could be defined, adding to the scope and applicability of our approach. We conclude with several more speculative applications of queries.

5.2. Virtual metrology prospects

We can establish that a physical artifact is within given specifications using physical metrology, verifying that the part is in conformance. In modern practice, specifications are a CAD model

[4] www.scan-and-solve.com.

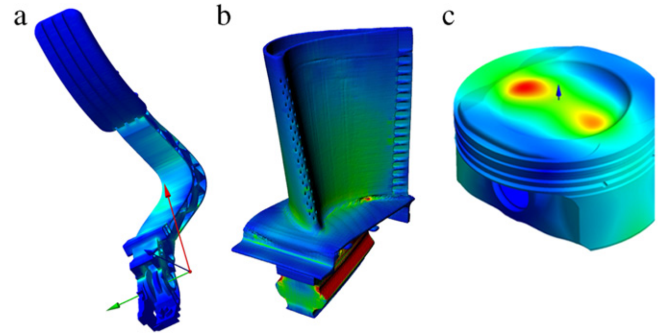


Fig. 4. Query based design/analysis integration is robust and tolerant: (a) successful analysis of a model that contains numerous geometric errors in its boundary representation (from [54]); (b) stress analysis of noisy reverse engineered turbine blade (courtesy of Intact Solutions); (c) piston model contains numerous small feature and complex surface topology (from [54]).

annotated with tolerance specifications. The represented net shape could contain errors, such as erroneous draft angles, improper gaps between features, etc.; see, e.g., [59]. By devising a suitable specification semantics and measurement queries to validate it, we can in principle establish whether an electronic CAD model, authored in a CAD system *C*, is within specification. In analogy to present metrological practice, the process establishing conformance would begin with a shape evaluation, such as sampling point measurements, and would complete the task with a shape comprehension computation that checks specific features. Certain queries might be added to the ones of Section 3 to streamline the process.

Before end users upgrade their CAD systems to the next major release, many perform due diligence and re-evaluate critical archived models to ensure that they remain usable. This is of particular importance for long-term archival. But how might one ensure that the re-evaluated model was correctly re-evaluated? Complex parts and large-scale assemblies necessitate an approach that is more rigorous than relying on visual inspection. Translation into STEP and comparing the resultant STEP models is unreliable: the STEP model may itself have errors, so if the two models diverge it could be the fault of the translator or of the CAD system. Virtual metrology may be an answer. For example, we could approximate the net shape by a polyhedral model, as discussed in Section 4.1, and compare whether the so evaluated models agree, within an acceptable tolerance. Some of the model deficiencies discussed in [59] can be revealed by a sufficiently detailed polygonization. Other deficiencies, such as flawed nonmanifold structures, can be exposed by specific queries designed to locate such flaws.

5.3. System integration

In addition to eliminating the bottleneck of meshing, the query-based implementation of analysis is easily parallelizable, tolerant and robust with respect to geometric errors, noise, and small features as illustrated in Fig. 4. CAD–CAE interoperability and integration can be done with queries, without understanding an opaque CAD representation or proprietary algorithms interpreting it. The same principles apply to other system integration tasks, not only in CAD/CAM, but also in consumer, biomedical, and cloud-based applications where geometric models are increasingly becoming important. Using queries, the information inherent in a data structure can be revealed without breaching proprietary walls or requiring disclosure of trade secrets. Moreover, when the functionality of the Service Oriented Architecture (SOA) interface is standardized, a task that partitions the standardization into small, tractable segments, software subsystems can be interchanged with ease.

Acknowledgments

The first two authors gratefully acknowledge support from the NIST under contract SB1341-12-SE-0587. Christoph Hoffmann has been supported in part by the NSF grant 0938999 and a gift from Intel. Vadim Shapiro's research was supported by the NSF grants CMMI-1029553 and CMMI-0856778.

The identification of any commercial product or trade name does not imply endorsement or recommendation by the National Institute of Standards and Technology.

References

[1] Hoffmann C, Shapiro V, Srinivasan V. Geometric interoperability for resilient manufacturing. Technical report CSD 11-015, Purdue. 2011.
[2] ISO-STEP part 42. Product data representation and exchange. Technical report, ISO. 1998.
[3] Smith Bradford, Brauner Kalman, Kennicott Philip, Liewald Michael, Wellington Joan. Initial graphics exchange specification(iges) version 2. 0. NTIS, SPRINGFIELD, VA(USA), 1983, 341, 1983.
[4] Yap Chee. Towards exact geometric computation. Computational Geometry: Theory and Applications 1997;7(1–2):3–23.
[5] Fortune S. Polyhedral modelling with exact arithmetic. In: Proc. 3rd ACM symp. solid modeling appl., 1995. p. 225–34.
[6] Keyser J, Krishnan S, Manocha D. Efficient and accurate B-rep generation of low degree sculptured solids using exact arithmetic. In: Proc. 4th ACM symp. solid modeling and appl. 1997. p. 42–55.
[7] Guibas L, Salesin D, Stolfi J. Epsilon geometry: building robust algorithms from imprecise computations. In: Proc. 5th ACM Symp. Comp. Geom. 1989. p. 208–17.
[8] Segal M. Using tolerances to guarantee valid polyhedral modeling results. In: Proc. ACM SIGGRAPH 90. 1990. p. 105–14.
[9] Jackson D. Boundary representation modelling with local tolerancing. In: 3rd ACM symp. solid modeling and appl. 1995. p. 247–53.
[10] Qi J, Shapiro V. Geometric interoperability with epsilon solidity. Journal of Computing and Information Science in Engineering 2006;6:213.
[11] Barequet G, Duncan C, Kumar S. RSVP: a geometric toolkit for controlled repair of solid models. IEEE TVCG 1998;4(2):162–77.
[12] Bøhn JH. Removing zero-volume parts from CAD models for layered manufacturing. IEEE CG&A 1995;15(6):27–34.
[13] Murali T, Funkhouser T. Consistent solid and boundary representations from arbitrary polygonal data. In: Proc. 1997 symp. interact. 3D graphics. 1997. p. 155–62.
[14] Sheng Xuejun, Meier R Ingo. Generating topological structures for surface models. IEEE CG&A 1995;15(6):35–41.
[15] Weihe Karsten, Willhalm Thomas. Reconstructing the topology of a CAD model—a discrete approach. Algorithmica 2000;26(1):126–47.
[16] Requicha AAG. Representations for rigid solids: theory, methods and systems. ACM Computing Surveys 1980;12(4):437–64.
[17] Braid I. Stepwise construction of polyhedra in geometric modelling. Mathematical Method in Computer Graphics and Design 1980;123–41.
[18] Edalat Abbas, Lieutier André. Foundation of a computable solid modelling. Theoretical Computer Science 2002;284(2):319–45.
[19] Qi J, Shapiro V. ε-topological formulation of tolerant solid modeling. Computer-Aided Design 2006;38(4):367–77.
[20] Spitz S, Rappoport A. Integrated feature-based and geometric CAD data exchange. In: Proc. 9th ACM symp. solid model. and appl. 2004. p. 183–90.
[21] Hoffmann CM, Juan R. Erep: an editable, high-level representation for geometric design and analysis. In: Peter RWilson, Michael JWozny, Michael JPratt, editors. Geometric modeling for product realization. Amsterdam: North-Holland; 1993. p. 129–64.
[22] Chen X, Hoffmann CM. Towards feature attachment. Computer-Aided Design 1995;27(9):695–702.
[23] Shapiro V, Vossler DL. What is a parametric family of solids? In: Proc. 3rd ACM symp. solid modeling and appl. 1995. p. 43–54.
[24] Raghothama S, Shapiro V. Boundary representation deformation in parametric solid modeling. ACM Transactions on Graphics 1998;17(4):259–86.
[25] Bidarra R. Validity maintenance in semantic feature modeling. Ph.D. thesis, Computer Graphics and CAD/CAM Group, TU Delft. 1999.
[26] Raghothama S, Shapiro V. Topological framework for part families. ASME JCISE 2002;2(4):246–55.
[27] O'Connor M, Rossignac J. SGC: a dimension independent model for pointsets with internal structures and incomplete boundaries. In: IFIP workshop geometric modeling. North-Holland; 1990.
[28] Shapiro V. Maintenance of geometric representations through space decompositions. International Journal of Computational Geometry and Applications 1997;7:383–418.
[29] Armstrong C. et al. Djinn: a geometric interface for solid modeling. In: Information geometers. 2000.
[30] Shapiro V. Solid modeling. In: Farin G, Hoschek J, Kim M-S, editors. Handbook of computer aided geometric design. Elsevier Science Publishers; 2002. p. 473–518.

[31] Shapiro V, Vossler DL. Construction and optimization of CSG representations. Computer-Aided Design 1991;23(1):4–20.
[32] Shapiro V, Vossler DL. Separation for boundary to CSG conversion. ACM Transactions on Graphics 1993;12(1):35–55.
[33] Ilies H, Shapiro V. The dual of sweep. Computer Aided Design 1999;31(3):185–201.
[34] Chen X, Hoffmann C. Design compilation for feature-based and constraint-based CAD. In: Proc 3rd ACM symp on solid modeling. 1995.
[35] Hounshell David. From the American system to mass production, 1800–1932: the development of manufacturing technology in the United States. JHU Press; 1985.
[36] Qi J, Shapiro V, Stewart NF. Single-set and class-of-sets semantics for geometric models. Computer-Aided Design 2006;38(10):1088–98.
[37] Shannon CE. Communications in the presence of noise. Proceedings of the Institute of Radio Engineers 1949;37:10–21.
[38] Hoffmann CM. Geometric and solid modeling. San Mateo (Cal.): Morgan Kaufmann; 1989.
[39] Chazal F, Lieutier A. Weak feature size and persistent homology: computing homology of solids in R^n from noisy data samples. In: Proc. 21st annl. symp. comp. geom. ACM; 2005. p. 255–62.
[40] Shapiro V. Real functions for representation of rigid solids. Computer-Aided Geometric Design 1994;11(2).
[41] Frisken S, et al. Adaptively sampled distance fields: a general representation of shape for computer graphics. In: 27th Annl.Conf. Comp. Graphics and interactive Techn. ACM; 2000. p. 249–54.
[42] Sethian JA. Level set methods: evolving interfaces in geometry, fluid mechanics, computer vision and material sciences. Cambridge University Press; 1996.
[43] Carlsson E, Carlsson G, De Silva V. An algebraic topological method for feature identification. IJCGA 2006;16(04):291–314.
[44] Tilove R. Set membership classification: a unified approach to geometric intersection problems. IEEE Transactions on Computers 1980;C-29:874–83.
[45] Kemmerer J Sharon. STEP: the grand experience. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, July 1999. NIST SP 939.
[46] Lorensen E William, Cline E Harvey. Marching cubes: a high resolution 3d surface construction algorithm. In: ACM siggraph computer graphics, vol. 21. ACM; 1987. p. 163–9.
[47] ASME. Y.14.5-2009. Dimensioning and tolerancing. 2009.
[48] Srinivasan V, Shakarji CM, Morse EP. On the enduring appeal of least-squares fitting in computational coordinate metrology. JCISE 2012;12: 011008–1–011008–15.
[49] Shapiro V, Tsukanov I, Grishin A. Geometric issues in computer aided design/computer aided engineering integration. Journal of Computing and Information Science in Engineering 2011;11:021005.
[50] Clark BW, editor. Proceedings of the 18th international meshing roundtable. Springer Verlag; 2009.
[51] Rvachev VL, Sheiko TI, Shapiro V, Tsukanov I. On completeness of RFM solution structures. Comp. Mechanics 2000;25:305–16.
[52] Höllig K. Finite element methods with B-splines. SIAM; 2003.
[53] Peskin CS. The immersed boundary method. Acta Numerica 2002;479–517.
[54] Freytag M, Shapiro V, Tsukanov I. Finite element analysis in situ. Finite Elements in Analysis and Design 2011;47:957–72.
[55] Kantorovich LV, Krylov VI. Approximate methods of higher analysis. Interscience Publishers; 1958.
[56] Rvachev VL, Sheiko TI, Shapiro V, Tsukanov I. Transfinite interpolation over implicitly defined sets. CAGD 2001;18:195–220.
[57] Freytag M, Shapiro V, Tsukanov I. Field modeling with sampled distances. Computer Aided Design 2006;38(2):87–100.
[58] Luft B, Shapiro V, Tsukanov I. Geometrically adaptive numerical integration. In: Proc. 2008 ACM symp. solid and phys. modeling. 2008. p. 147–57.
[59] International Techne Group. Identifying and resolving model quality problems. Slide Deck, 2000. Communicated by V. Srinivasan in 2011.