Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2009), *23*, 197–214. Printed in the USA.
Copyright © 2009 Cambridge University Press 0890-0604/09 $25.00
doi:10.1017/S0890060409000134

Reusability-based selection of parametric finite element analysis models

NSIKAN UDOYEN AND DAVID W. ROSEN

Systems Realization Laboratory, George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA

(RECEIVED September 23, 2007; ACCEPTED August 18, 2008)

Abstract

A selection method to support adaptive reuse of parametric finite element analysis (FEA) models is introduced in this paper. Adaptive reuse of engineering artifacts such as FEA models is common in product design, but difficult to automate because of the need to integrate new information. The proposed method factors reusability into selection by evaluating models based on comparative estimates of effort involved in adapting them for reuse to model a query problem. The method is developed for FEA models of component-based designs. FEA modeling of electronic chip packages is used to illustrate the method's usefulness. We conclude with a discussion on the method's advantages and limitations and highlight important issues for further research.

Keywords: Engineering Information Management; Finite Element Analysis; Model Reuse; Reusability

1. INTRODUCTION

Adaptive reuse of parametric finite element analysis (FEA) models involves iterative integration of new information into a *reference model* to solve a new problem. *Parametric* FEA models are programmatic implementations of FEA models. A *reference model* is an archived model that is updated to solve the new physical problem. In adaptive reuse of a parametric FEA model, commands used to programmatically implement the reference model, rather than data inputs alone, need to be updated to reuse the reference model to solve a new physical problem.

An aim in selecting a suitable reference model is to reduce the amount of adaptation needed to reuse the modeling strategy. Selection of a reference model that describes a slight variation of the new problem reduces the amount of searches for missing information. Integrating new information into a reference model that incorporates assumptions consistent with the new problem and the modeler's intent will require less changes to individual tasks in the modeling strategy. The *modeler's intent* is defined here to encompass

- the physical problem the modeler is trying to solve and
- assumptions made by the modeler to help formulate the problem, which may include geometric simplifications, or assumptions about behavioral properties of structural components in the design.

Information that helps a modeler select a reference model is usually found in a model's supporting documents. This includes sources such as computer-aided design (CAD) models, data sheets, scholarly publications, sketches, notes, or even comments in the model script. These sources help explain and justify assumptions made in creating the reference model, or simply document data transformations that allow product data to be used in the model. The need to select based on the contents of heterogeneous collections of related documents makes manual selection time consuming.

FEA modeling of *component-based designs* is one area that would benefit from automated selection of reference models based on reusability-oriented concerns such as required adaptation. Component-based designs consist of several connected structural components and are configured from catalogs of structural components. The incremental redesign that occurs when such designs are reconfigured to meet new specifications leads to adaptive reuse of FEA models to validate new design parameters. Common adaptive reuse scenarios occur when models are reused to solve new problems that

- are variations of existing problems that include new structural components or use different materials and
- subject the device to a different loading condition, causing a new failure mode to be of interest.

Reprint requests to: David W. Rosen, Systems Realization Laboratory, George W. Woodruff School of Mechanical Engineering, Manufacturing Research Center, Room 252, Georgia Institute of Technology, 813 Ferst Drive, NW, Atlanta, GA 30332-0405, USA. E-mail: david.rosen@me.gatech.edu

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

As an example of an adaptive reuse scenario, consider the selection from reference model options to develop a FEA model to assess solder joint fatigue in an underfilled *flip chip on board* (FCOB) chip package. Flip-chip technology, illustrated in Figure 1, is one of several approaches to electrically and mechanically connecting a silicon chip to a printed circuit board, also known as direct chip attachment methods (Coombs, 1996). The mismatch in coefficients of thermal expansion between the bonded parts causes stresses and strains to develop in the structure when it is subjected to thermal loading. This results in physical phenomena such as solder joint fatigue, die cracking, mask delamination, and substrate board warpage that are responsible for various failure modes.

Because simple analytical models are usually insufficient to assess the stated failure modes, FEA is often used as part of the assessment of the reliability of the overall product. Most of the failure modes in question cannot be directly assessed from FEA models. FEA is used to estimate exogenous properties like stress and strain, which are then used in empirical models to assess the failure modes (Lee et al., 2000). The use of FEA to help accurately assess these failure modes for different package types and new materials is an active research area (Qu & Yao, 1999; Gustafsson et al., 2000; Hanna et al., 2000; Pang & Chong, 2001).

Two potential reference models are shown. They are both highly simplified two-dimensional models of FCOB assemblies under thermal loading conditions. The first option, shown in Figure 2, was developed to measure warpage in a similar package. It has the same structural components, but inadequate geometric detail if the modeler's objective is to assess solder joint fatigue. A geometric *lumping* assumption was used to represent underdie material (solder joint and underfill) as a single geometric region with aggregated prop-

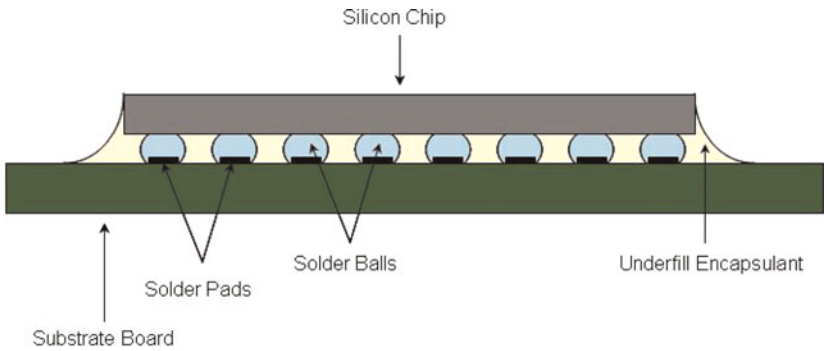


Fig. 1. A cross-section view of a flip chip on board chip package. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

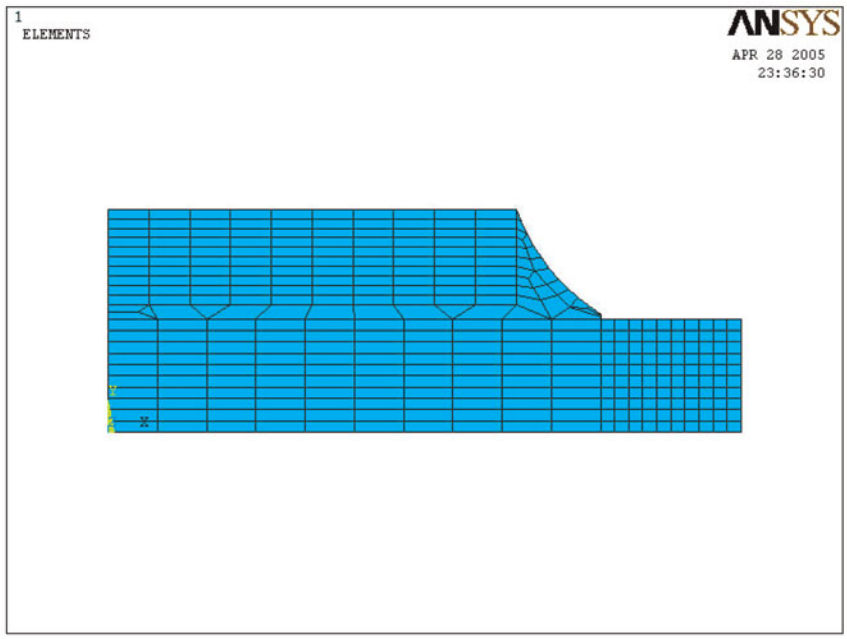


Fig. 2. A two-dimensional FEA model of an underfilled flip chip on board with lumping assumption. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

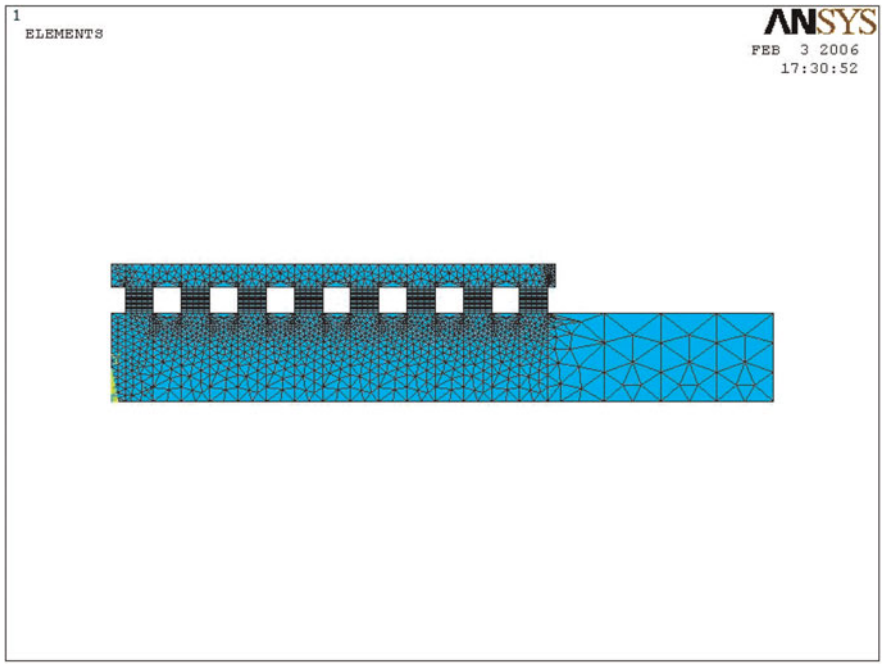


Fig. 3. A two-dimensional FEA model of a flip chip on board without underfill. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

erties. The second option, shown in Figure 3, was developed to measure solder joint fatigue in a similar structure without underfill. The level of detail is thus adequate but reuse will involve retrieval, modeling, and integration of properties of the underfill.

Concerns about the effort involved in adapting each of the reference models to suit the modeler's intent inform the modeler's perception of the *reusability* of the modeling strategies they implement. Selection of a reference model for adaptive reuse involves comparing relative reusability of several models. The central aim of this research is to develop a method that incorporates reusability assessments into automated retrieval of FEA models for adaptive reuse. The central research question addressed in this paper is thus "How can reusability toward a modeler's intent be incorporated into automated selection of FEA models for adaptive reuse?"

An automated selection method to support adaptive reuse of FEA models of component-based designs is presented in this paper. We propose the use of declarative, attributed, metadata representations of FEA model creation processes as a means of supporting automated reasoning about reusability of FEA models. The representations are also used to index heterogeneous collections of information associated with FEA models. Related research is discussed in Section 2. A conceptual data model for FEA modeling process description is introduced in Section 3. The method is introduced in Section 4. Examples are used to illustrate the method's usefulness in Section 5. Section 6 consists of a discussion of the work done that highlights important features and observations. Significant contributions are summarized in Section 7.

2. RELATED RESEARCH

2.1. Automating routine reuse

Research on reuse of FEA models emphasizes automating *routine* reuse of FEA models, or raising the level of discourse between the user and the application. *Routine* reuse involves using new inputs in existing models or anticipated variations of the problems they model. Several researchers have successfully applied knowledge-based inference methods toward automation of routine reuse for narrowly defined sets of problems representing specific application domains of the finite element method (FEM; Finn, 1993; Shephard & Wentorf, 1994; Fenves & Turkiyyah, 1996). Support for automated routine reuse of FEA models has also been achieved by modeling *associativity* between design and analysis model representations (Peak, 1993; Peak et al., 1999).

Adaptive reuse scenarios are difficult to fully automate using rule-based systems. Unanticipated structural changes may cause changes to modeling strategies that force automation rules to be updated. Missing information may also prompt searches for information and the need to apply new techniques to integrate such information into the reference model (Hanna et al., 2000). The resulting manual adaptation of reference models often involved in adaptive reuse scenarios makes automated support more feasible for specific reuse tasks such as retrieval, rather than the entire adaptive reuse process.

2.2. Data access methods

Data access methods are a class of approaches developed to make results of archived FEA models easily accessible.

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

An object-oriented data access system for FEA models was developed by Peng et al. (2003). The system enables queries for results of FEA models and can support queries on nonlinear analysis models. Results are retrieved from a relational database or regenerated. A *class system* to represent data structures of finite elements was introduced by Mackie (1999). A data model distributes FEA model data around a graphical model by associating it with object representations of graphical entities (nodes, lines, etc.), which are arranged hierarchically, rather than storing such data centrally in arrays. The solution is shown to help control data access and localize the effect of changes to the FEA model.

Representations developed to support data access methods enable association of data with graphical entities. They are not semantically rich enough to support selection of a reference model for adaptive reuse. Considerations such as consistency of assumptions with a modeler's intent cannot be automatically inferred from data access representations. The impact of missing data on the reuse effort must also be factored into the selection of the model before adaptation. The scope of information represented must be expanded to enable automated selection based on these considerations.

2.3. Model retrieval

In a review of case-based retrieval methods, Azuaje et al. (2000) classify case-based retrieval methods based on the techniques used to infer relevance, as shown in Figure 4. Their approach leads to two broad classes of case-based retrieval methods, referred to as *indexing-based* and *distance-based* methods. A third class of approaches called *hybrid* methods is proposed for methods that use aspects of both indexing-based and distance-based methods to infer relevance.

Indexing-based methods are defined as methods that require creation of *indexing structures* to represent and organize cases to be retrieved, and support automated reasoning about their relevance. Distance-based methods are defined as methods that rely on computation of *semantic distances* between case representations. Searches are then performed to return the *nearest k* neighbors or cases that fall below a set semantic distance threshold. Semantic distances are computed using mathematical operations on case representations.

Feature vector representations of cases are commonly used to support semantic distance computation. This allows semantic distance computation to be reduced to simple vector operations.

Plate (2000) identifies two approaches to feature vector creation:

- *Localized feature vectors:* Each digit in this vector representation represents a term in a case description. A localized feature vector can be *binary*, with each term indicating the presence or absence of a specific term in the case description, or *weighted*, with each digit reflecting the relative importance of the term in the case description.
- *Distributed feature vectors:* Vectors are used to represent disjoint primitive concepts in a vocabulary of terms used to describe items being retrieved. More complex terms and case descriptions are created by combining vectors using operations such as vector addition or convolution.

Research on automated retrieval of FEA models has also resulted in several representation and retrieval approaches (Nagasawa et al., 1995; Fujio, 2003; Udoyen & Rosen, 2006). Nagasawa et al. (1995) presented a case-based approach to

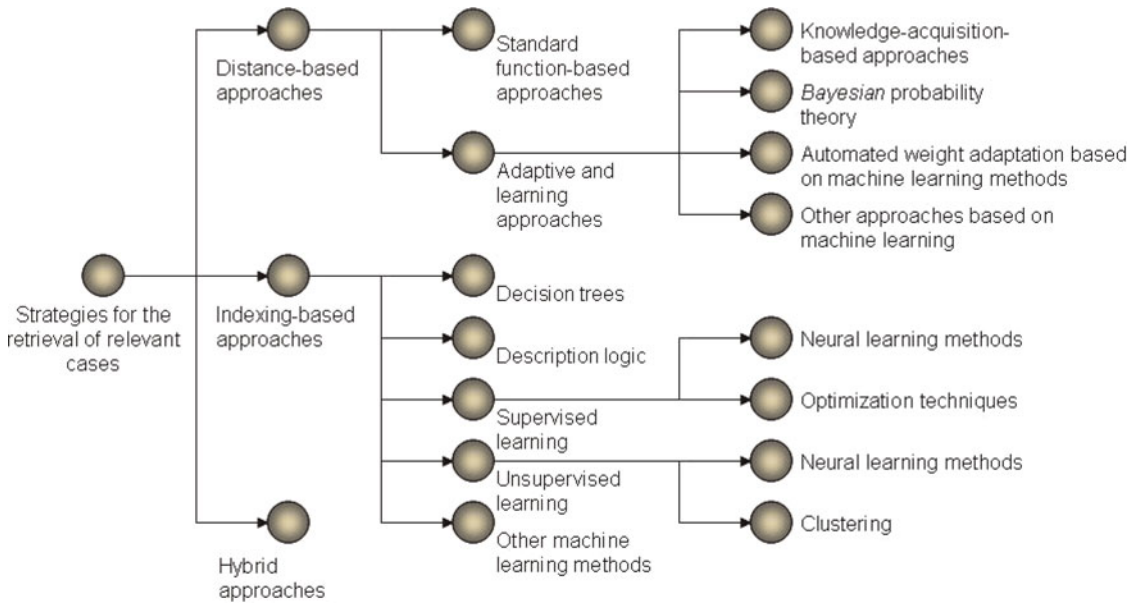


Fig. 4. Case-based retrieval strategies (Azuaje et al., 2000). [A color version of this figure can be viewed online at journals.cambridge.org/aie]

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

retrieving FEA models that uses a cosine function as a quantitative measure of similarity between representations of FEA models. Localized vector representations of FEA models are generated based on binary vector representations of data inputs. The digits in the binary vectors reflect terms in a vocabulary of input parameters. Fujio provided an FEM repository system that acts as a code generator and provides users with a high-level programming interface for creating models (Fujio, 2003). The interface is based on a scripting language consisting of several *functional blocks* used to describe FEM data objects.

The method proposed by Fujio also focuses on FEM data, much like the data access methods. The use of a localized vector-based representation precludes the description of complex relationships that exist between data in FEA models represented by vector contents. Such relationships include *part-of* relationships and other dependencies between information that arise during the model creation process. A modeler's understanding of these relationships and dependencies allows him to assess the relative costs of reusing reference models.

2.4. Model representation

Several researchers have also attempted to develop more complete representations of analysis models including FEA models. Grosse et al. (2005) proposed a set of ontologies for classifying engineering analysis modeling knowledge. Their ontology is designed to describe concepts that form the basis of engineering analysis models. The authors note that existing modeling approaches fail to address issues that lead to poor reusability (routine reuse), adaptation (adaptive reuse), and interoperability. Missing features that hinder adaptation are listed as

- the ability to capture nature and intent of adaptation and
- the ability to combine or sequence adaptations and recognize incompatibilities in them.

Chabanas and Promayon (2004) propose a unified representation of physical models designed to cover approaches used to simulate soft human tissue. Their approach is based on XML and supports development of general modeling architectures for their problem space. Their language consists of two languages, Loads Modeling Language (LML) and Physical Modeling Language (PML), defined in XML Schema to describe geometry, *part-of* hierarchies, loads, constraints, and wrappers to solvers.

2.5. Reusability

Approaches to developing reusability-oriented retrieval methods have been outlined in research on software components (Basili & Rombach, 1991). The problem of software component selection is similar in that artifacts consisting of executable code need to be retrieved and ranked based on properties not explicitly represented in the artifacts or automatically inferable directly from their content. Reusability metrics have been developed for ranking and classification of software components (Freeman & Prieto-Diaz, 1987). Inoue et al. (2005) introduce a reusability-based ranking method based on directed graph representation of the component library. Etzkorn et al. (2001) present an approach to evaluating object-oriented software based on quality factors related to reusability.

Research on reusability has not translated into automated support for adaptive reuse of engineering artifacts. To provide automated support for effective engineering information management we must consider factors such as heterogeneity (Kashyap & Sheth, 2000) and complex relationships that exist between engineering information (McMahon et al., 1995; Nelson et al., 1996; Pearson, 1996). We propose to address these factors through the development of semantically rich representations of FEA models and domain-independent reasoning services. Metadata-based approaches have been used to support retrieval of heterogeneous collections of information (Shklar et al., 1995). Automated metadata extraction methods often used in these approaches are not feasible when information includes formal (FEA model script, CAD model) and informal data sources (research publication, sketches, notes, comments).

The use of ontologies to structure metadata representations of FEA models and associated information is the basis of our approach. Our proposed ontology was developed based on conceptualizations of the FEA model creation process designed to support routine reuse (Arabshahi & Barton, 1991; Fenves & Turkiyyah, 1996). Inference methods supported by this ontology were also developed as part of our approach, to incorporate general reusability factors for models of component-based designs into the reusability assessment. The metadata indexes created using our ontology are high-level process models that describe creation of FEA models and relationships and dependencies between tasks and information used and generated in the process. We show the use of such process models to support domain-independent reasoning about reusability of FEA models of electronic chip packages to illustrate the method's potential usefulness for FEA models of component-based designs.

3. CONCEPTUAL DATA MODEL

The selection method is designed to assess relative reusability of reference models based on tasks and information needed for adaptive reuse. The information required to automate this comparison is not captured explicitly in FEA model scripts. The method thus relies on a manually created representation of the model creation process to support automated reasoning about reusability.

The conceptual data model for the representations is based on the conceptualization of the FEA model preprocessing shown in Figure 5. This conceptualization is based on published work on automated support for FEA model preprocessing (Arabshahi & Barton, 1991; Finn, 1993; Peak, 1993;

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

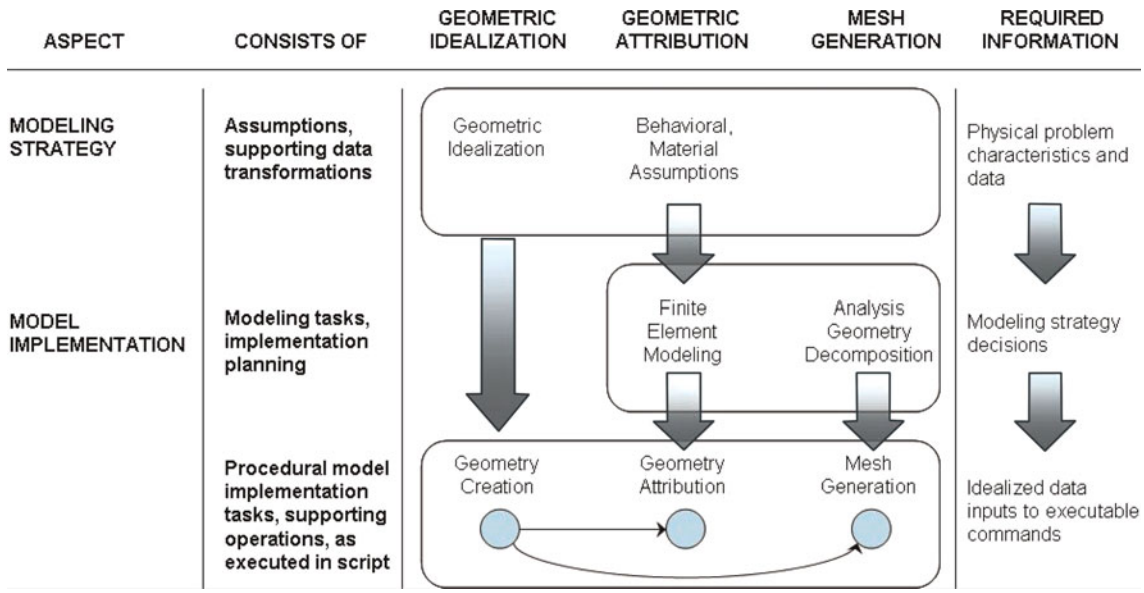


Fig. 5. FEA model preprocessing. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

Fenves & Turkiyyah, 1996). FEA model preprocessing encompasses the steps involved in formulating and encoding a physical problem for solution by a FEA solver. FEA model preprocessing steps use and generate formal and informal design information. The process model in Figure 5 represents a domain-independent view of the FEA model creation process that emphasizes the types of tasks involved and how they support each other (blue arrows). The process is thus described as a set of interdependent types of tasks, which are executed as information is made available, rather than as a linear sequence of tasks.

FEA model creation also involves reasoning at different levels of abstraction to support different aspects of the FEA model creation process, generally classified (Arabshahi & Barton, 1991) as the following:

- idealization: simplification of the physical problem's geometric or behavioral properties to facilitate modeling
- attribution: association of idealized properties to geometric domains in the analysis geometry
- mesh generation: discretization of the analysis geometry for solution using the FEM.

The conceptualization presented in Figure 5 considers two perspectives on the model creation process that are essential to reasoning about reusability of FEA model implementation scripts. These perspectives are reflected in the two types of process information described in the conceptual data model:

1. Modeling strategy: reasoning and modeling decisions that impact model implementation, including assumptions made, data, transformations, and behavioral models used as a result of these decisions
2. Model implementation:
   a. procedural implementation of the FEA model and supporting finite element modeling tasks
   b. *part-of* view of idealized analysis geometry
      - reflects the modeler's perspective on the physical problem described by the model, how the modeler decomposes the problem to reason about it
      - captures the mapping between structural components in the physical problem and the analysis geometry, reflects the level of detail at which the problem is modeled

The conceptual data model consists of declarative entities with string attributes, which are summarized in tables in the next few subsections. These entities are used to create historical representations of FEA model implementations that serve as indexes to heterogeneous collections of documents that include FEA model implementation scripts and information used to create them. An example of a modeling process description created from these entities can be found in Appendix A.

3.1. Modeling strategy description entities

Modeling strategy description entities describe informal modeling activities that are not captured explicitly in the procedural implementation of the FEA model, but determine how and why specific physical problem data is used in the model. The two types of modeling strategy description entities are shown in Table 1.

Assumption entities describe the simplifying assumptions made in a modeling process and the aspects of the model's

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

Table 1. *Summary of modeling strategy description entities*

Entity	Attributes	Attribute Types
Assumption	Name	String
	Subject	Component or material property or load
	Supports	Implementation task
	Creates	Geometric domain (lumping assumptions only)
	Removes	Component (lumping, feature removal, substitution only)
	Negates	Assumption (optional)
	Requires	Assumption (optional)
Transformation	Name	String
	Creates	Parameter
	Removes	Components

implementation code affected. The assumption's impact on the model's implementation code is described through the dependencies that exist between the assumption entities and other modeling process description entities. The representation of assumptions developed by Fenves and Turkiyyah (1996) modeled an assumption as an entity consisting of declarative and procedural components. The representation used here is fully declarative, as our objective is not to automate reuse, but to describe how assumptions influence the model's implementation code to enable reasoning about the reusability of a model script. The commands used to implement specific tasks and the inputs often reflect a combination of several modeling assumptions. An example would be the simultaneous use of symmetry, geometric lumping, and dimensional reduction in simplifying the analysis geometry for the model shown in Figure 2.

The assumption's relationship to other aspects of the model's implementation is described through the assumption entity's attributes. For geometric simplification assumptions such as feature removal, dimensional reduction, and feature substitution, the analysis model entities created and removed because of the application of an assumption are listed under the *creates* and *removes* attributes. Any other assumptions that must be applied in conjunction with the assumption or that cannot exist in a problem at the same time are listed under the *negates* attribute.

Transformation entities describe instances in which device data are transformed to create inputs to executable statements used to implement the model. The data inputs, parameters created by the transformation, and process steps and conditions (modeling tasks, assumptions) that made the transformation necessary or possible are stated as the entity's attributes. Transformation entities do not describe *how* data is transformed. Rather they help show that these transformations were completed as part of the modeling strategy. By listing the data inputs and outputs, they indicate what must be known about a physical problem for certain tasks to be possible if the strategy associated with a model is to be reused.

3.2. Model implementation description entities

Model implementation description entities are used to describe the procedural implementation of the model as a set of implementation tasks and operations that support these tasks. A set of declarative entities is used to describe the dependencies between parts of the implementation and the modeling process. Such relationships need to be captured if the impact of model adaptation actions such as relaxing assumptions on the model's implementation and strategy are to be automatically assessed. Model implementation description entities are described in Table 2.

Geometric domain entities describe the mapping between the structural elements in the physical problem and the analysis geometry. This mapping allows us to automatically determine whether a model is detailed enough, by asserting that a specific structural element is explicitly represented. The analysis geometry is described as a *part-of* hierarchy, as shown in Figure 6. The decomposed analysis geometry shown describes an underfilled FCOB with a geometric lumping assumption. The structural components referenced by the lumped geometry are listed under the *lumped entity* node.

Operation entities describe fragments of the procedural code used to implement specific geometry creation, attribution or mesh generation tasks for individual structural components or aggregations of structural components. Through their attributes, they indicate the following:

- the motivation behind code fragments, that is, the implementation task they support;
- the physical problem data they use;
- the physical problem component or geometric domain in the analysis model affected by the implementation operation; and
- any modeling tasks that must be performed prior to the operation.

By describing how specific code fragments are related to the reasoning behind the model and the physical problem data, the impact of adaptation actions can be assessed from

Table 2. *Summary of model implementation description entities*

Entity	Attributes	Attribute Types
Operation	Supports	Implementation task
	Subject	Geometric domain
	Inputs	Parameter
	Requires	Modeling task (optional)
Implementation task	Name	String
Modeling task	Name	String
	Subject	Component or material property or load
Geometric domain	Name	String
	References	Component(s)
	HasParent	Geometric domain or analysis geometry

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

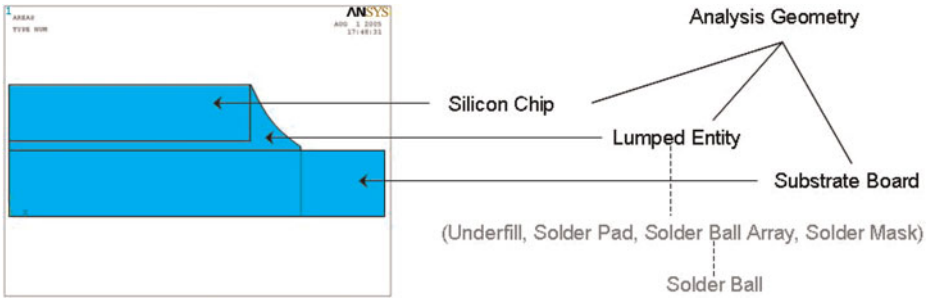


Fig. 6. An analysis geometry representation for the FEA model of a flip chip on board with lumping assumption. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

a detailed process-oriented perspective, enabling the modeler to answer questions such as *what parts of implementation tasks need to be redone? What parts of the code need to be updated? What data is needed?*

Implementation task entities represent the three broad classes of FEA preprocessing tasks implemented using executable commands in a FEA model script: geometry creation, geometry attribution, and mesh generation. An implementation task is considered *complete* if it is implemented for every geometric domain in the analysis geometry.

Modeling task entities describe activities that must be carried out before specific operations are possible, such as `Decompose Substrate Board Geometry`. Such activities can be very time consuming and contribute significantly to the effort involved in adapting a model. They may also be dependent on information generated by other tasks. Associating them with specific modeling operations further elaborates the cost of manually adapting a specific part of the model. Modeling tasks often require that a modeler reference a source for tips on how to carry out the task and what to consider. As such, these entities also point to sources referenced to guide the modeling task in the development of the reference model.

4. SELECTION METHOD

The conceptual data model supports automation of several evaluations critical to a modeler's qualitative estimation of the *effort* involved in adaptive reuse of a FEA model. Such estimation is important in selecting a reference model, or even determining whether adaptive reuse is an appropriate reuse mode. A flowchart for the automated evaluation of individual reference models is shown in Figure 7. The evaluation rates reusability based on automated projection of tasks needed to adapt the reference model to the new problem. The automated evaluation method is enabled by several components, namely, assessment routines, adaptation routines, and reusability metrics.

Automated evaluation of reference models is executed by comparing a query expression of a modeler's intent to a

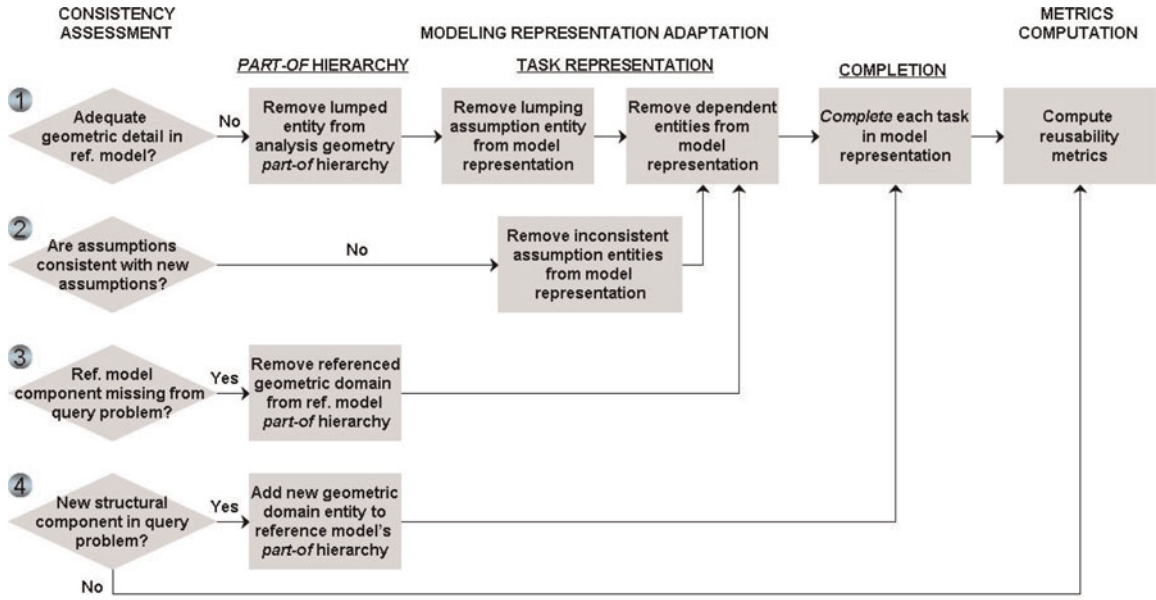


Fig. 7. An automated evaluation of a reference model based on a modeler's intended reuse. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

representation of each reference model. The modeler's intent is expressed as a declarative entity with three attributes:

1. Part-of hierarchy
   - This is a partonomic description of the component-based design being analyzed.
2. Component of interest
   - This is the component whose exogenous property is used to assess the phenomenon of interest.
3. Modeling assumptions
   - This describes any assumptions made by the modeler.

The results of the automated evaluation are compiled into a metric that enables comparative evaluation of reference models. The coarseness of the task-based conceptual data model allows the evaluation method to remain domain independent. Individual components of the model evaluation method are described in detail in the next few sections.

4.1. Assessment routines

4.1.1. Consistency

The first type of assessment concerns the *internal consistency* of the model, when constraints introduced by the modeler are considered. In reasoning about consistency, it is assumed that the modeler has already determined the specific phenomenon of interest to assess using the model. The model must contain adequate geometric detail to support evaluation of the phenomenon of interest. The modeler may also have decided on specific modeling assumptions to simplify the FEA model. Newly introduced assumptions must also be able to combine with those already made in the reference model and describe a consistent analytical system. Inconsistencies in either area prompt reimplementation of affected model creation tasks. Understanding the impact of inconsistencies thus enables projection of the extent of adaptation required.

The evaluation currently uses four types of consistency assessments for component-based designs, as shown in Figure 7.

1. Is the model detailed enough? The model must be detailed enough to enable assessment of the phenomenon of interest. Adequacy of geometric detail is inferred by determining that the structural component whose exogenous property (stress, strain, displacement, etc.) is needed to measure the phenomenon of interest is explicitly represented in the analysis geometry.
2. Are there new structural components in the query problem?
3. Does the reference model problem have components that are not in the query problem?
4. Are assumptions in the reference models consistent with the modeler's assumptions?

4.1.2. Completeness

The second type of assessment supported by the conceptual data model concerns the *completeness* of model implementation. Procedural implementation of a model is *complete* when each attribution and mesh generation task has been implemented for all geometric domains in the analysis geometry. The concept of completeness allows a modeler to reason about which new or updated tasks are needed to adapt a model to a new problem.

A *complete* procedural implementation of a FEA model of a FCOB is shown in Figure 8. Exploded views of some implementation tasks are shown. The modeling process described is based on the *part-of* view of the analysis geometry shown in Figure 6. The *part-of* view of that analysis geometry is shown in the boxes under each task in Figure 8. The circles at nodes in the hierarchy represent operations on structural components or aggregations of structural components in the model's implementation code. For example, the task `Assign Finite Element` is implemented by a total of three operations. Two of these operations attribute geometric domains that reference individual structural components. The third operation attributes the geometric domain that references an aggregation of underlie structural components. The task `Free Mesh Geometry` consists of a single operation that results in the entire analysis geometry being free meshed.

Model implementation task incompleteness is inferred in a domain-independent manner by comparing the hierarchical representation of the analysis geometry to the model implementation description task. Completeness of model implementation tasks is inferred using a general completeness criterion abstracted from conceptual data model representations of FEA models. The completeness criterion reads as follows:

> . . . a task is complete if every leaf node in the part-of representation or its ancestor is the subject of an operation in support of the task.

Model implementation descriptions are complete if all of their constituent implementation tasks are complete. An assumption here is that because of the connectivity between structural components, loads and boundary conditions are system properties, even when applied directly to individual components.

4.2. Adaptation routines

Adaptation routines identify operations in a modeling process that need to be added, updated, or removed to reuse a FEA model for a new problem. Adaptation routines manipulate model implementation descriptions in response to the discovery of any of the four types of inconsistencies listed in

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

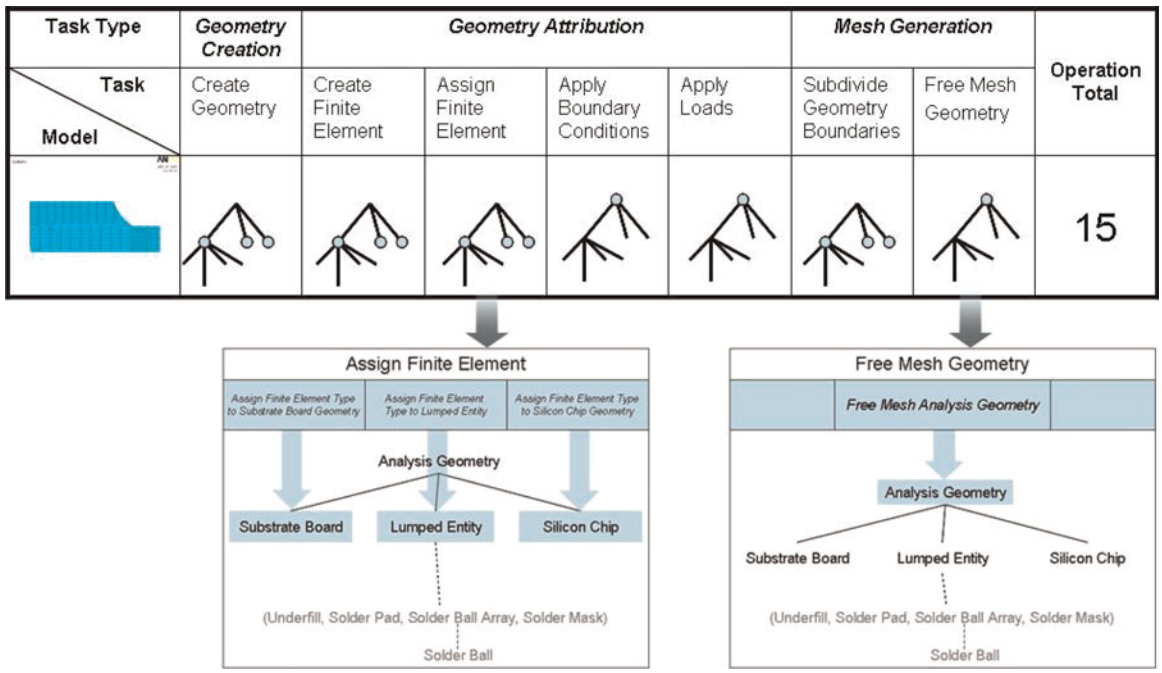


Fig. 8. The complete implementation of a two-dimensional FEA model of a flip chip on board chip package. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

Section 4.1.1. Generally, these routines automatically remove inconsistent entities from the model implementation description, and then add operations needed to complete the altered modeling process description.

Different types of inconsistencies that prompt adaptation routines can result from the reuse of a single model. In each case, the model completeness criterion is used to determine whether suggested adaptations result in a complete implementation. Information generated through the use of adaptation routines is used to compute a reusability metric. The adaptation routines are described briefly.

1. *Addition of a structural element:* Whenever a query contains a structural element not referenced in the archived model being evaluated, the hierarchical *part-of* view of the analysis geometry is updated. A new geometric domain entity is created to represent the new structural component. Operations are added to the model implementation description until the completeness criterion is satisfied.
2. *Lumping assumption relaxation:* It is assumed that the lack of sufficient geometric detail is due to the use of a lumping assumption on aggregated components. To relax a lumping assumption, the lumping assumption entity is removed from the modeling process description. The lumped geometric domain created by the assumption is also removed from the *part-of* hierarchical view of the analysis geometry. New geometric domains are then created for components referenced by the lumped entity and inserted in the *part-of* hierarchy. Model implementation and strategy entities that require the removed assumption entity are removed from the FEA model description. New operations are then added to the model implementation description until the completeness criterion is satisfied.
3. *New assumption is introduced:* To apply a new assumption to a reference model, any assumptions that are inconsistent with it must be removed along with dependent process entities. New operations are then added to the model implementation description until the completeness criterion is satisfied.
4. *Structural element in reference model is not in query problem:* If the reference model describes a component-based design with structural components not used in the query problem, the geometric domain entities referencing the unused components are removed from the hierarchical view of the analysis geometry. Note that this view lists the *types* of components present, and not every instance of each component type. Modeling implementation and strategy entities that depend on the removed entities are removed, along with their dependent entities. New operations are then added to the model implementation description until the completeness criterion is satisfied.

4.3. Reusability metric

A metric is used to compare reference models based on the results of the reusability assessments. The metric compares reference models based on an assessment of

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

- the effort involved in adapting the model and
- the impact of missing data on reusability.

The effort involved in adapting a reference model is included by an *adaptability* metric. Adaptability is modeled as

$$\frac{\text{number of procedural operations that need to be added or updated to adapt reference model}}{\text{number of operations in projected adaptation}} \quad (1)$$

Lower adaptability metric values describe reference models that require less adaptation. The adaptability metric favors reference models with more detailed strategies that require less adaptation.

The impact of missing data on reusability is approximated by summing factors that would motivate searches for missing data. This component of the metric is referred to as *data deficiency* and includes as components

- the number of operations that require data from new structural components, referred to here as *newCompOperations*, and
- the number of new structural components, referred to here as *newComponents.*

These factors are incorporated into the metric as

$$dataDeficiency = (a \times (count(newCompOperations))^2) + (b \times (count(newComponents))) \quad (2)$$

where a and b are constants with default values of 0.001 and 0.1. The component for *newCompOperations* is squared because dependencies often emerge between new operations that imply new modeling tasks, data transformations, and possibly new assumptions.

The two aspects of reusability computed using Eqs. (1) and (2) are combined in a reusability metric. The metric is computed as

$$reusability = (k \times adaptability) + dataDeficiency \quad (3)$$

where k is a constant with a default value of 0.1. The metric enables comparison of reference models, and has a minimum value of 0. Reference models with lower values are ranked higher. The constant values have been selected to make the reusability metric value a small fraction for the most desirable reference models.

For this example, default values of a, k, and b were selected to weight the metric in favor of FEA models of designs with the most similar structural components, based on the assumption that supporting documents stored with the model would contain more relevant information to aid reuse. Default values for the coefficients were set as follows:

1. The coefficient b was set to 0.1 to make the number of new structural components the dominant component of the metric.
2. The coefficient a was intentionally given a smaller value (0.001) to keep its contribution to the metric low in comparison to the number of new structural components. The component *newCompOperations* was squared to reflect the cost of integrating new information into a model. Procedural implementation of new operations is usually preceded by informal modeling tasks, such as modeling of material properties, and searches for

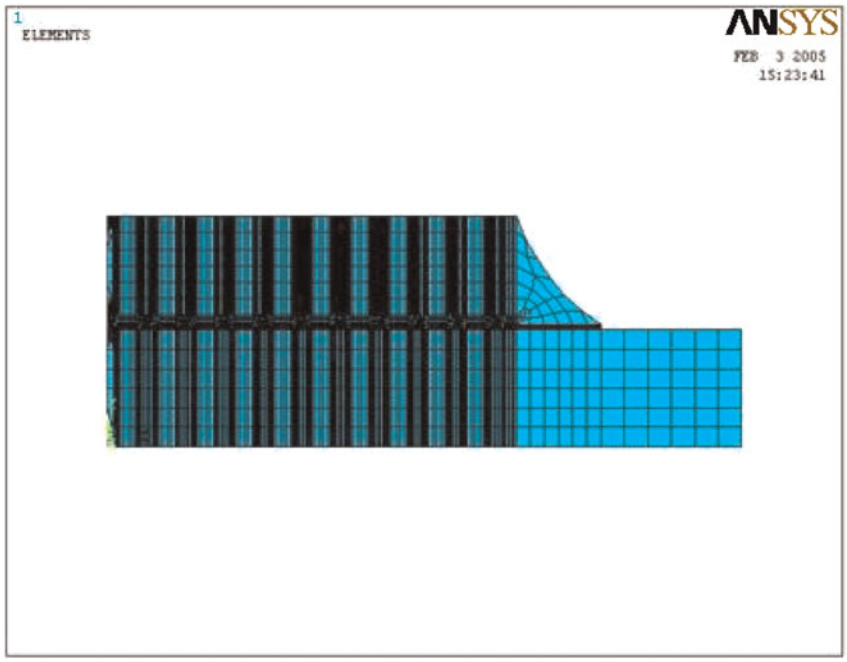


Fig. 9. The adapted FEA model. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

data. For example, relaxing a geometric lumping assumption may require recalculation of properties for individual structural components if such information is not explicitly captured in the reference model.

The number and difficulty of supporting tasks does not always vary proportionally or linearly with the number of new operations because implementation of operations on other geometric domains may be affected. For example, insertion of new geometric domains may cause the modeler to subdivide geometric domains representing adjacent connected structures differently to facilitate meshing, as can be seen by the differences in the mesh for the substrate board for different analysis geometries in Figures 2, 3, and 9. This component of the metric is thus squared to reflect the potential increase in complexity as more information needs to be integrated into a model.

3. The *adaptability* component is already a fraction, so setting the coefficient k to 0.1 ensures that its contribution to the metric remains small relative to the contribution from the number of new components.

In all cases constant values can be adjusted based on empirical results to reflect relative importance of these aspects of reusability.

5. EXAMPLES

A framework was developed to implement the selection method. The conceptual data model and supporting routines were implemented in CLIPS™ (Riley, 2005). Metric computation routines were written in Java and were able to read evaluation results directly from the CLIPS inference engine using a Java-based interface called jCLIPS (Menken, 2005).

A sample query was run on two different sets of conceptual data model representations of several reference models. In the first case, the objective was to illustrate the use of the method to select from models of very similar component-based designs with different modeling assumptions. For the second example, the query was executed again, on a field of reference model problems with a wider range of structural similarity to the query problem. The objective was to examine the selection results when default coefficient values are used, and suggest potential improvements to the method.

In the first example, the two models introduced in Section 1 were evaluated as options for adaptive reuse to assess solder joint fatigue in an underfilled FCOB. The modeler's intent was expressed declaratively as follows:

```
Intent:
       Structure:
              Components:
                     epoxyUnderfill
                     siliconChip
                     solderBallArraySn63Pb37
                     solderMask
                     solderPad
                     substrateBoard
              Loads:
                     timeVaryingThermalLoad
                            subject: Structure
       Component of Interest:
              solderBallArraySn63Pb37
       Assumptions:
              [Symmetry, Dimensional Reduction]
```

Both reference models would need to be adapted for reuse. The extent of adaptation required for either FEA model is illustrated in Figure 10. The FEA model shown in Figure 9 is consistent with the modeler's simplifying assumptions and has adequate geometric detail to support assessment of the phenomenon of interest.

Required adaptations to either reference model can be seen from the graphical process model representations in Figure 10. Operations in the implementation of Model 1 are shown in the first row. The second row describes the operations in the implementation of Model 2. The projected adaptation of either modeling strategy to model the structure described in the query is captured in the third row, based on new operations that would be needed.

In each row, the *part-of* view of the analysis geometry shown beneath each task indicates which structural components are represented by geometric domains in the model. The *part-of* view in the first row in Figure 10 refers to the view shown earlier in Figure 6. The second and third rows represent models that do not use the lumping assumption for underdie materials, so the structural components in the FCOB are all represented in the analysis geometry. The adaptation of models 1 and 2 is discussed in further detail in the following sections.

5.1. Model 1

The first reference model describes an underfilled FCOB under a thermal load. It was developed to assess substrate board warpage in the same component the modeler intends to analyze. The physical problem had the same structural components as the problem to be modeled, made of the same materials. The modeling strategy used in this reference model applied a geometric lumping assumption to the underdie components.

The projected adaptation required to apply the reference model's implementation strategy to the modeler's intent is summarized in Figure 10. The need to fulfill consistency and completeness requirements for the modeler's intent forced the following suggested adaptations:

1. *Inconsistency: component of interest not explicitly represented*
   a. The lumped geometry was removed from the *part-of* description of the analysis geometry so that the solder ball array could be represented explicitly.
   b. All operations on the lumped entity were removed from the process description. (*−4 operations*)

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

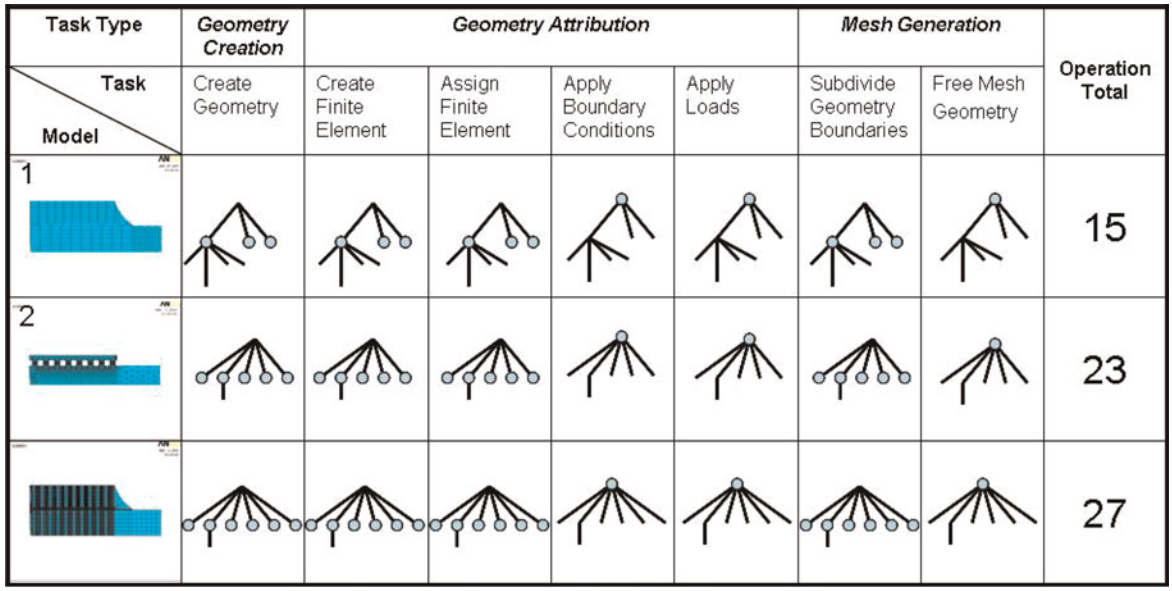


Fig. 10. The projected adaptation of the reference models. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

c. Operations dependent on the removed operations were iteratively removed from the process description. (*−4 operations*)

2. *Adaptation*

   a. Distinct geometric domains were created for the structural components initially referenced by the lumped entity. This forced the *part-of* hierarchy to change, as shown in the task cells for the adapted model in Figure 10. Four new nodes were added in place of the node removed in step 1a.
   b. Operations were added for the new domains in support of all tasks made *incomplete* by step 2a. (*+16 operations*)
   c. Operations were added for the tasks made incomplete by step 1c. (*+4 operations*)

Adaptability was calculated as 20/(15 + 12), or 0.7407, whereas *data deficiency* was

$$(0.001 \times 0^2) + (0.1 \times 0) = 0,$$

because no new components are considered in the modeling process. Using the default weightings, the overall relevance value based on process-oriented metrics was computed as

$$0.07407 + 0 = \underline{0.07407.}$$

5.2. Model 2

The second model was developed to assess solder joint fatigue in the same type of chip package without underfill. The projected adaptation required to apply the reference model's implementation strategy to the modeler's intent is summarized in Figure 10. The following suggested adaptations in computing the projected adaptation were automatically inferred:

1. *Inconsistency: new structural component*
2. *Adaptation*

   a. A geometric domain was created for the underfill, which is the new structural component. The *part-of* description of the analysis geometry was updated to include a node for the underfill, as shown in the task cells for the adapted model in Figure 10.
   b. Operations were added for the new domains in support of all tasks made *incomplete* by step 2a. (*+4 operations*)

Adaptability was calculated as 4/(23 + 4), or 0.1481, whereas *data deficiency* was

$$(0.001 \times 4^2) + (0.1 \times 1) = 0.116.$$

Using the default weightings, the overall relevance value based on process-oriented metrics was computed as

$$0.01481 + 0.116 = \underline{0.13081.}$$

The smaller reusability metric value for the first model implied that it was more relevant for the modeler's intended reuse. This is in agreement with the observation that it is based on a problem that is structurally more similar to the query problem than the second model. If this particular result is considered undesirable, the weightings of metric components can be adjusted as needed.

In the second example, the field of reference models was expanded. The repository was updated with representations of two-dimensional FEA models of the chip package types

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

shown in Figure 11 under thermal loading conditions. The modeling strategies described for each package type did not incorporate any lumping assumptions, or behavioral assumptions inconsistent with the query problem. A model of the underfilled FCOB without the lumping assumption was also included in the repository. The packages were also assumed to be modeled at a similar level of detail. Some entities shown in the schematic diagrams in Figure 11 such as vias, were assumed to be omitted from the models because their presence has little effect on relevant failure modes.

A sampling of selection results is shown in Table 3 for reference models in which common structural components in the query problem and reference models (e.g., solder ball arrays) were made of the same material. These would represent the most similar instance in each class of models. The ranking closely reflects the number of similar structural components each package type shares with the query problem because of the emphasis placed on this component of the reusability metric. However, the method enables selection between models of the same structure, as can be seen from the metric value for the underfilled FCOB without the lumping assumption. This is a significant improvement over selection methods based on structural similarity alone.

The high rating given to the FCBGA model is interesting because the FCBGA is a significantly more complex structure than the FCOB, with several additional structural components. However, it incorporates the same components used in the underfilled FCOB, in the same arrangement. The high rating reflects the assumption that relevant information needed to model the query problem would be used in this reference model. The rating also reflects the lack of dependencies between operations removed in the course of adapting the model, and operations required to implement the FEA model, because of the modeling strategy used. The historical representation of the model did not indicate that the removal of unwanted operations would cause the required operations to be updated. However, the cost of removing the extra operations from the procedural implementation of the model was not considered in computing the metric, and it is not always insignificant.

6. DISCUSSION

In this section, we outline several issues and observations concerning the development and application of the selection method. In discussing these issues we hope to outline limitations and advantages of the method, while suggesting areas for further research.

6.1. Simplifications

The reusability metric is a comparative measure of usefulness based on the projected *amount* of activity and information needed to adapt archived models to support a modeler's intended reuse. In developing the selection method, emphasis was placed on reducing the amount of time spent searching for missing information and manually integrating it into a

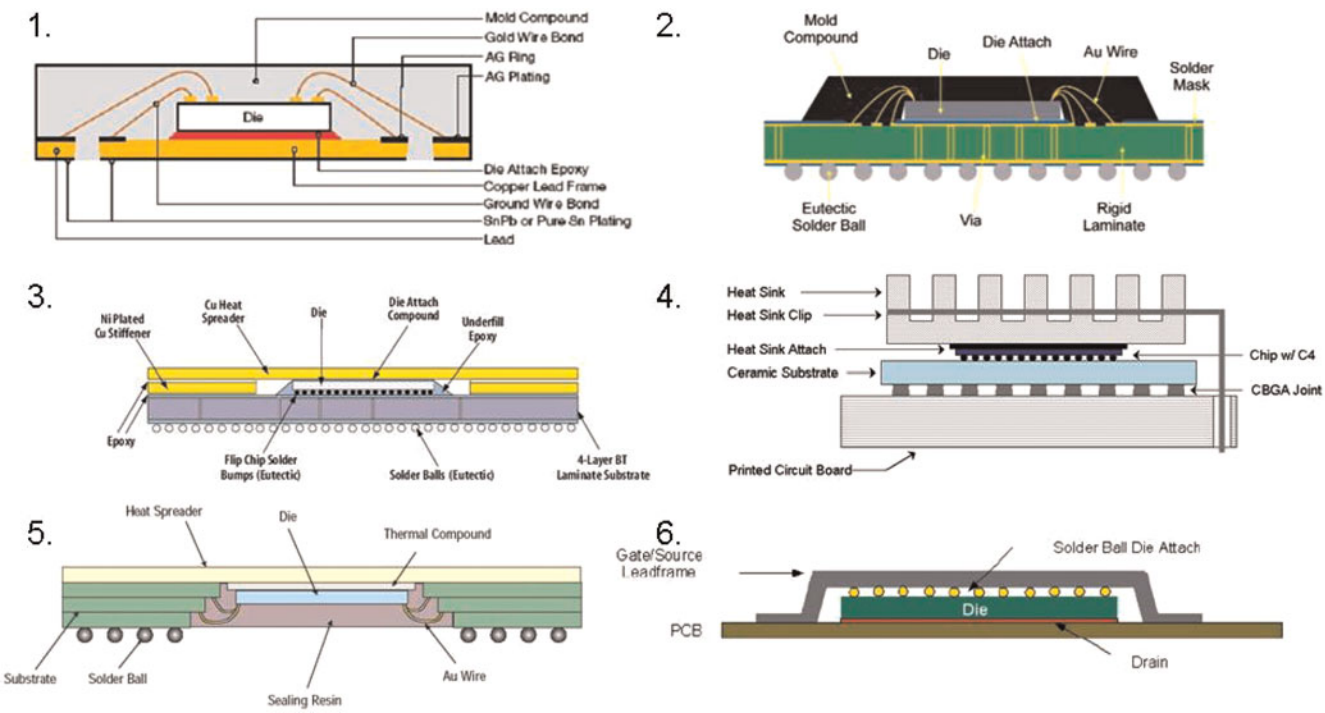


Fig. 11. The repository contents: 1. leadless plastic chip carrier (ASAT, 2006), 2. plastic ball-grid array (Amkor, 2006), 3. flip chip ball-grid array (LSI Logic, 2006), 4. ceramic ball-grid array (Deletage et al., 1996; Pang et al., 2000), 5. enhanced ball-grid array (Fujitsu, 2006), and 6. flip chip lead molded package (Yong et al., 2002; Fairchild Semiconductor, 2006). [A color version of this figure can be viewed online at journals.cambridge.org/aie]

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

Table 3. *Reusability metric values*

Package Type	Reusability
Underfilled FCOB	0.000
FCBGA	0.000
PBGA	0.231
EBGA	0.343
FLMP	0.346
CBGA	0.346
LPCC	0.463

reference model. To keep the selection method objective, several simplifications were made:

1. In attempting to estimate the impact of dependencies between operations on reuse, differences between types of dependencies between operations were ignored. The removal of an operation may result in removal or varying degrees of adaptation of dependent operations, possibly even the introduction of new assumptions.
2. The effort involved in adapting specific operations is based on several subjective factors, such as the modeler's expertise, and availability of data, and was therefore not considered. Operations were therefore given equal weight in determining reusability.
3. In ranking the heterogeneous collections of documents retrieved, subjective factors such as the quality of supporting documentation were ignored. Reference models are reused by modelers with varying degrees of skill. There is no reliable measure of the usefulness of particular documents. Some modelers may be able to integrate data found in data sheets or scholarly publications, while others may need to see code fragments or other supporting information from similar models that demonstrate how retrieved information is used in a model.

6.2. Domain independence

The use of inference procedures that relied on operations as the unit of information to describe strategies enabled the development of algorithms that operate independently of the domain of application of the FEM. The extensibility of the selection method implied by this is significant, because the FEM is widely used in various engineering domains. The key here is the use of general properties of the FEA preprocessing process as implemented by FEA model scripts, such as *completeness* and *consistency*, which can be formally inferred from conceptual data models for FEA models of component-based designs.

6.3. Representation creation

Manual creation of metadata representations of models used in this method could be very time consuming. There is also a possibility of errors that could impact metric calculation. The *part-of* hierarchy used in query description relied on a controlled vocabulary of unique structural component names. The modeler's awareness of the scope of the vocabulary could limit his ability to effectively create model representations, especially when dealing with large repositories. User interface design is one suggested approach to addressing these problems and limiting associated errors.

6.4. Integration of the selection method into a retrieval method

The method outlined in this paper supports selection of reference FEA models, which is one aspect of their semantic retrieval for adaptive reuse. To be useful, it needs to be integrated with a method that supports retrieval by efficiently excluding less relevant models. Other semantic retrieval tasks may require different conceptual data models or augmentation of the conceptual data model presented here to support their automation. Automated abstraction of conceptual data model entities from semantically rich product models or finite element representations would also be helpful.

6.5. Application development

Although the reusability metric may be useful as a comparative measure, other inferences supported by the conceptual data model could be just as useful. A modeler may want to look beyond the metric to select from models with very similar metric values. In such cases, a list of tasks needed for adaptation of each reference model may be more useful to the modeler. Such information is inferred in the process of compiling the metric and could help the modeler make strategic decisions such as the reuse mode. For example, the CBGA, EBGA, and FLMP returned similar metric values. However, all three models represent designs that are significantly different from the query problem. In all cases, time will be spent retrieving missing data. Depending on the modeling tasks that need to be redone, a modeler may be better off developing a new model, using the reference model to guide the implementation of specific tasks, and possibly reusing small fragments of the model's implementation code.

6.6. Model accuracy as a selection criterion

A goal of this research was the development of semantic retrieval methods for FEA models that incorporated aspects of the modeler's intent. In their work on taxonomies for engineering analysis models, Grosse and colleagues (2005) capture *accuracy expectation* as an aspect of intent. Although this property can be used to describe models that are routinely reused, *a priori* reasoning about accuracy in adaptive reuse is significantly more complex because new information is introduced. A modeler may introduce new assumptions based on his qualitative understanding of their impact on the model's prediction of the phenomenon of interest. For example, in two-dimensional thermomechanical modeling of electronic

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

chip packages, use of the plane strain assumption to compensate for modeling the structure as a thin strip may cause an overestimation of the stiffness in the overall structure. This introduces error in exogenous properties such as deflection, strain, and stress, which are used to assess failure modes. Without accounting for qualitative physics, and possibly incorporating problem domain-specific reasoning, it will be difficult to support consistent automated reasoning about the impact of new information on a model's accuracy.

7. CONCLUSION

A reusability-oriented selection method for parametric FEA models has been introduced in this paper. The method uses a metadata representation to enable evaluation of FEA models for adaptive reuse. The method is intended for FEA models of component-based designs and its use has been illustrated in the selection of FEA models of electronic chip packages. The use of a conceptual data model to structure the metadata representations allows important information not explicitly captured in computer-interpretable form to be captured and reasoned about automatically. Although the process of creating the representations can be labor intensive and error prone, the inclusion of modeling process information has been shown to enable more precise selection than structural similarity alone.

The applicability of the method to FEA models of component-based designs is inferred from the domain independence of the conceptual data model and supporting inferences. The development of the method highlights research issues and difficulties in incorporating reusability into automated selection of FEA models for adaptive reuse. It is hoped that further research on these issues will enable the development of automated reusability-based selection methods applicable to other types of engineering artifacts, and general approaches to automated reasoning about reusability of engineering artifacts.

ACKNOWLEDGMENTS

We gratefully acknowledge the support for the first author from the George W. Woodruff School of Mechanical Engineering during his PhD studies. We also thank Krishna Tunga and Dr. R. Pucha of the Computer Aided Simulation of Packaging Reliability (CASPaR) Lab at Georgia Institute of Technology for their help in understanding simulation of electronic chip packages.

REFERENCES

Amkor. (2006). *PBGA data sheet.* Accessed February 26, 2006, at http://www.amkor.com/products/alldatasheets/PBGA.pdf

Arabshahi, S., & Barton, D.C. (1991). Towards integrated design and analysis. *Finite Elements in Analysis and Design 9*, 271–293.

ASAT. (2006). *LPCC application notes.* Accessed February 27, 2006, at http://www.asat.com/products/data/appsnotes/lpcc_ apps_04.pdf

Azuaje, F., Dubitzky, W., Black, N., & Adamson, K. (2000). Retrieval strategies for case-based reasoning: a categorized bibliography. *The Knowledge Engineering Review 15*, 371–379.

Basili, V.R., & Rombach, H.D. (1991). Support for comprehensive reuse. *Software Engineering Journal 6(5)*, 303–316.

Chabanas, M., & Promayon, E. (2004). Physical modeling language: towards a unified representation for continuous and discrete models. *Int. Symp. Medical Simulation*, pp. 256–266.

Coombs, C.F. (1996). *Printed Circuit Handbook*. New York: McGraw–Hill.

Deletage, J., Fenech, A., Bechou, L., Ousten, Y., Danto, Y., Salagoity, M., Faure, C., & Rao, S. (1996). Thermomechanical behavior of ceramic ball grid array based on experiments and FEM stimulations. *IEEE/CPMT Int. Electronics Manufacturing Symp*., pp. 91–98.

Etzkorn, L.H., Hughes, W.E., & Davis, C.G. (2001). Automated reusability quality analysis of OO legacy software. *Information and Software Technology 43*, 295–308.

Fairchild Semiconductor. (2006). *FLMP design guide.* Accessed February 27, 2006, at http://www.fairchildsemi.com/ collateral/flmp_designguide.pdf

Fenves, S.J., & Turkiyyah, G. (1996). Knowledge-based assistance for finite element modeling. *IEEE Expert 11*, 23–32.

Finn, D.P. (1993). A physical modeling assistant for the preliminary stages of finite element analysis. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 7*, 275–286.

Freeman, P., & Prieto-Diaz, R. (1987). Classifying software for reusability. *IEEE Software 4(1)*, 6–16.

Fujio, H. (2003). The Feelfem system: a repository system for the finite element method. *Int. Parallel and Distributed Processing Symp*., p. 254. Nice: IEEE.

Fujitsu. (2006). *EBGA.* Accessed February 27, 2006, at http://www.fujitsu.com/downloads/MICRO/fma/pdf/enchancedpkg.pdf

Grosse, I.R., Milton-Benoit, J.M., & Wileden, J.C. (2005). Ontologies for supporting engineering analysis models. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *19*,1–18.

Gustafsson, G., Guven, I., Kradinov, V., & Madenci, E. (2000). Finite element modeling of BGA packages for life prediction. *IEEE Electronic Components and Technology Conf*., pp. 1059–1063.

Hanna, C., Raghunathan, R., & Sitaraman, S. (2000). Development of virtual reliability methodology for area-array devices used in implantable and automotive applications. *IEEE Transactions on Components and Packaging Technologies 23(3)*, 452–461.

Inoue, K., Yokomori, R., Yamamoto, T., Matsushita, M., & Kusumoto, S. (2005). Ranking significance of software components based on use (relations. *IEEE Transactions on Software Engineering 31(3)*, 213–225.

Kashyap, V., & Sheth, A. (2000). *Information Brokering Across Heterogenous Digital Data: A Metadata-Based Approach*. Boston: Kluwer Academic.

Lee, W.W., Nguyen, L.T., & Selvaduray, G.S. (2000). Solder joint fatigue models: review and applicability to chip scale packages. *Microelectronics Reliability 40*, 231–244.

LSI Logic. (2006). *Flip chip ball grid array 4-layer (FPBGA-4L) package family.* Accessed May 10, 2006, at http://www.lsilogic.com/files/docs/marketing_docs/asic/FPBGA-4Lps.pdf

Mackie, R.I. (1999). Object-oriented finite element programming—the importance of data modeling. *Advances in Engineering Software 30*, 775–782.

McMahon, C.A., Pitt, D.J., Yang, Y., & Sims Williams, J.H. (1995). An information management system for informal design data. *Engineering with Computers 11*, 123–135.

Menken, M. *jCLIPS*. Accessed November 22, 2005, at http://www.cs.vu.nl/~mrmenken/jclips/

Nagasawa, S., Nomura, M., Miyata, Y., & Sakuta, H. (1995). Development of a retrieval system for case data of finite element modeling. *Int. Computers in Engineering Conf. Database Management Symp*., pp. 245–253. Boston: ASME.

Nelson, P.R., Poltrock, S.E., & Schuler, D. (1996). Industrial strength hypermedia: managing engineering information with hypermedia. *SIGOIS Bulletin 17(2)*, 18–33.

Pang, J.H.L., & Chong, D.Y.R. (2001). Flip chip on board solder joint reliability analysis using 2-D and 3-D FEA models. *IEEE Transactions on Advanced Packaging 24(4)*, 499–506.

Pang, J.H.L., Wang, Z.P., & Seetoh, C.W. (2000). CBGA solder joint reliability evaluation based on elastic-plastic-creep analysis. *Journal of Electronic Packaging 122*, 255–261.

Peak, R.S. (1993). *Product model-based analysis models: a new representation of engineering analysis models*. PhD Thesis. Georgia Institute of Technology.

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

Peak, R.S., Scholand, A., Tamburini, D., & Fulton, R.E. (1999). Towards the routinization of engineering analysis to support product design. *International Journal of Computer Applications in Technology 12(1)*, 1–15.
Pearson, S. (1996). Economical management of engineering information. *ISA Transactions 35(1)*, 3–8.
Peng, J., Liu, D., & Law, K.H. (2003). An engineering data access system for a finite element program. *Advances in Engineering Software 34*, 163–181.
Plate, T.O. (2000). Analogy retrieval and processing with distributed vector representations. *Expert Systems 17*, 29–40.
Qu, Y., & Yao, Q. (1999). Three-dimensional versus two-dimensional finite element modeling of flip-chip packages. *Journal of Electronic Packaging 121*, 196–201.
Riley, G. (2005). *CLIPS*. Accessed November 22, 2005, at http://www.ghg.net/clips/CLIPS.html
Shephard, M.S., & Wentorf, R. (1994). Toward the implementation of automated analysis idealization control. *Applied Numerical Mathematics 14*, 105–124.
Shklar, L., Sheth, A., Kashyap, V., & Shah, K. (1995). InfoHarness: use of automatically generated metadata for search and retrieval of heterogenous information. *Proc. 7th Int. Conf. CAiSE'95 on Advanced Information Systems Engineering* (Iivari, J., Lyytinen, K., & Rossi, M., Eds.), Vol. 932, pp. 217–230. Jyväskylä, Finland: Springer–Verlag. Accessed at citeseer.ist.psu.edu/article/shklar95 infoharness.html
Udoyen, N. (2006). *Information modeling for intent-based retrieval of finite element analysis models*. PhD Thesis. Georgia Institute of Technology.
Udoyen, N., & Rosen, D.W. (2006). Description logic representation of finite element analysis models for automated retrieval. *ASME Int. Design Engineering Technical Conf., Computers and Information in Engineering Conf*., pp. 603–616, Paper No. DETC2006-99451. Philadelphia, PA: ASME.
Yong, L., Irving, S., Tumulak, M., & Cabahug, A.E. (2002). Assembly process induced stress analysis for new FLMP package by 3D FEA. *IEEE Electronic Components and Technology Conf*., pp. 604–610.

David W. Rosen is a Professor and Associate Chair for graduate studies in the School of Mechanical Engineering at the Georgia Institute of Technology. He received his PhD at the University of Massachusetts in 1992 and his masters and bachelors degrees from the University of Minnesota in 1987 and 1985, respectively, all in mechanical engineering. He is a Fellow of ASME and is a former Chair of the ASME Computers and Information in Engineering Division. Dr. Rosen has published 9 book chapters, more than 50 journal articles, and over 150 conference papers. His professional activities include additive manufacturing and design methodology.

Nsikan Udoyen is a Design Engineer at Intel. He received his PhD from Georgia Institute of Technology in 2006, his MS from Stanford University in 1999, and his BS from Prairie View A&M University in 1998, all in mechanical engineering. Dr. Udoyen's research interests consist primarily of computer-aided engineering topics, encompassing engineering design, design and analysis integration, and information modeling.

APPENDIX A: MODELING PROCESS DESCRIPTIONS

A.1. FCOB with no underfill

This is a sample metadata representation for a procedural implementation of a FEA model. This example does not capture all of the information that would be used and generated in these tasks, but is sufficiently detailed to describe dependencies that exist between operations implemented in the model.

A.1.1. Implementation tasks

These entities list the model preprocessing tasks executed by the code in the ANSYS scripts for this model.

```
Implementation Task
     name: Render Geometry
     subject: Analysis Geometry
Implementation Task
     name: Create Finite Elements
     subject: Analysis Geometry
Implementation Task
     name: Assign Finite Elements
     subject: Analysis Geometry
Implementation Task
     name: Generate Mesh
     subject: Analysis Geometry
Implementation Task
     name: Apply Loads
     subject: Analysis Geometry
```

A.1.2. Geometric domains

The *geometric domain* entities describe the mapping between geometric domains in the analysis geometry and structural elements in the physical problem. All structural components are represented explicitly, so they map to geometric domains in the model.

```
Geometric Domain
     name: siliconChipGeometry
     references: siliconChip
     hasParent: Analysis Geometry
Geometric Domain
     name: substrateBoardGeometry
     references: substrateBoard
     hasParent: Analysis Geometry
Geometric Domain
     name: solderBallArrayGeometry
     references: solderBallArray
     hasParent: Analysis Geometry
Geometric Domain
     name: solderMaskGeometry
     references: solderMask
     hasParent: Analysis Geometry
Geometric Domain
     name: solderPadGeometry
     references: solderPad
     hasParent: Analysis Geometry
```

A.1.3. Modeling tasks

These entities list the modeling tasks that precede model implementation. They help model dependencies between these tasks and specific parts of the model's implementation, as some of these tasks may be required to generate inputs for some operations. In this case, a research publication was consulted to derive temperature-dependent properties of the solder used.

```
Modeling Task
     name: Model Temperature-dependent
     Material Properties for Sn62Pb36Ag2
```

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press

```
      subject: solderBallArray
      source: D:/papers/Qu99
```

A.1.4. Assumptions

The assumptions used in the modeling process are listed below.

```
Assumption
      name: Symmetry:
      subject: Structure
      relevance: Render Geometry
Assumption
      name: Dimensional Reduction
      subject: Structure
      relevance: Render Geometry
```

A.1.5. Transformations

Sample data transformations are listed below.

```
Transformation
      name: Solder Ball Width
      inputs: solderPad.width
      creates: SolderBallArrayGeometry.
      solderBallWidth
Transformation
      name: Substrate Board Width
      requires: Symmetry
      inputs: substrateBoard.width
      creates: substrateBoardGeometry.width
Transformation
      name: Silicon Chip Width
      requires: Symmetry
      inputs: siliconChip.width
      creates: siliconChipGeometry.width
```

A.1.6. Operations

A representative sample of the operation entities in this FEA model description is listed below. The operations describe procedures implemented as executable statements in the model script. The operations in support of the task Render Geometry are shown below. A more complete description of this and other FEA model descriptions can be found in a related publication (Udoyen, 2006).

```
Operation
      name: Render Substrate Board
      subject: substrateBoardGeometry
      supports: Render Geometry
      inputs:[substrateBoard.height,substrate BoardGeometry.width]
Operation
      name: Render Solder Ball Array
      subject: solderBallArrayGeometry
      supports: Render Geometry
      inputs:[solderBallArrayGeometry.solder BallHeight,
                solderBallArrayGeometry.solderBall Width]
      requires: Solder Ball Width
Operation
      name: Render Solder Mask
      subject: solderMaskGeometry
      supports: Render Geometry
      inputs:[solderMaskGeometry.height, solderMaskGeometry.width]
Operation
      name: Render Solder Pad
      subject: solderPadGeometry
      supports: Render Geometry
      inputs:[solderPadGeometry.height, solderPadGeometry.width]
Operation
      name: Render Silicon Chip
      subject: siliconChipGeometry
      supports: Render Geometry
      inputs:[siliconChip.height,siliconChip Geometry.width]
```

https://doi.org/10.1017/S0890060409000134 Published online by Cambridge University Press