6

Integration of a Domain Conceptual Model with the EDIF Information Model

Rachel Y.W. Lau and Hilary J. Kahn
Department of Computer Science
University of Manchester
Manchester M13 9PL
UK
email: rachel@cs.man.ac.uk, hilary@cs.man.ac.uk

1 Background

In the area of electronic design, EDIF (Electronic Design Interchange Format) is a well accepted standard for design exchange between different CAD systems. Information Modelling has been widely used by the EDIF community to describe the design information carried by an EDIF file. The EDIF Version 3 0 0 Information Model [1] represents formally the semantics of EDIF Version 3 0 0 and is the basis from which the EDIF Version 3 0 0 syntax [2] has been derived. In addition, the model provides the basis for the future extension of EDIF. Currently, work is in hand to extend EDIF into other domains such as PCB (Printed Circuit Board) and Test. The relevant EDIF Technical Sub-Committees were therefore asked to develop their domain conceptual models as a basis for integration into the core EDIF model in the future. Their domain conceptual models specify a set of requirements which should be met by future versions of EDIF. At present, the modelling activity is focused on creating an integrated model between the EDIF Version 3 0 0 Information Model and the PCB Conceptual Model [3] developed by the EDIF PCB Technical Sub-Committee.

This paper addresses several issues related to model integration, in particular on large-scale models. In the case of the core EDIF model and the PCB domain model, both models have hundreds of objects defined in their domain of interest. Hence, it is important to divide this large task into several phases. Figure 1 shows the process of integration between the PCB Conceptual Model and the EDIF Information Model.

Firstly, the PCB Conceptual Model undergoes a translation process so that it uses the same notation as in the core EDIF model. The model integration process is roughly divided into two phases. The first phase is to achieve the model integration between the core model and the PCB model at the conceptual level. The target of this phase is to establish possible mappings between PCB objects and EDIF objects. This phase requires active participation from the EDIF PCB Technical Sub-Committee (EDIF PCB TSC) and also the EDIF Technical Experts Group (EDIF TEG). The EDIF PCB TSC defines the scope of EDIF-PCB integration and also provides expert advice on the PCB domain. The EDIF TEG offers advice on EDIF issues which help to maintain conformity and consistency in any extension of EDIF. In many cases, there is more than one way to map the PCB objects into EDIF. Therefore experts from both committees need to decide which mapping should be used for the final integrated model. The second phase creates the fully integrated model based on the mappings identified in the first phase. In addition, it ensures coherence and consistency in the final integrated model.

F. J. Rammig et al. (eds.), *Electronic Design Automation Frameworks*

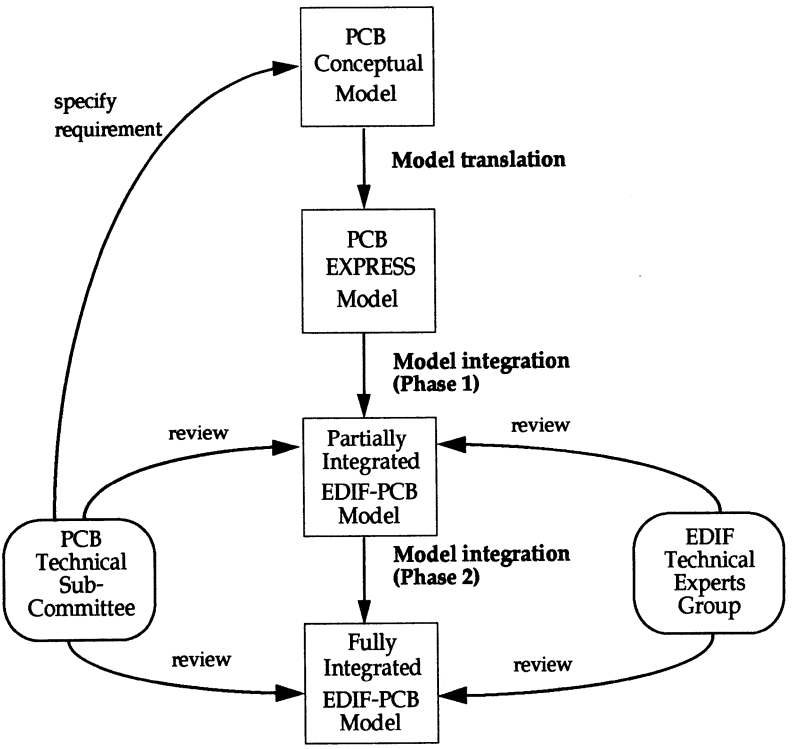

Figure 1: Integration process between the PCB Conceptual Model and the EDIF Information Model

2 Model translation

The first task of the integration is to make sure that the core EDIF model and the domain conceptual model use the same notation. The PCB Conceptual Model is used as the model integration example here. The PCB Conceptual Model is specified in both textual and graphical form. Each object defined in the PCB Conceptual Model is given a plain English textual description, together with a detailed list of attributes and the associated constraints. In contrast, the core EDIF model is specified formally in EXPRESS [4] - a modelling language which describes a domain in terms of schemas, entities and constraints. An EXPRESS schema defines a universe of discourse in which the defined objects have a related meaning and purpose. An EXPRESS entity defines an object of interest and is described by a number of attributes. An entity may have rules which constrain the value of individual attributes or a combination of attributes for every entity occurrence. In order to facilitate comparison between the formal core model and the semi-formal domain model, the PCB Conceptual Model is translated into EXPRESS so that it shares the same notation as in the EDIF model. Each PCB object is converted into an EXPRESS entity. The major advantage of this translation process is that the domain model will be formally defined and subsequently can be checked by existing EXPRESS checkers for correctness. Hence, the translation process makes sure that the domain model is self-consistent and unambiguous,

before any integration process takes place.

The translation of PCB objects into EXPRESS is a fairly straightforward task. However, additional information is added to the PCB EXPRESS model. One useful aspect involved is to determine whether an attribute of an object indicates an ownership relationship or a reference relationship. This information is useful for deciding the top level entities in the PCB model - an essential phase in mapping between the PCB objects and EDIF objects. This is done by careful examination of attribute names and their supporting documentation. In the PCB Conceptual Model, if an attribute uses terms such as "has" or "owns", it may imply an ownership relationship. For example, a *pin_group* object has an attribute *has_pins* which is a set of *pins*. It implies a *pin_group* owns a set of *pins*. On the other hand, if an attribute uses terms such as "refers" or "is_made_of", it may imply a reference relationship. For example, a *physical_layer* object has an attribute *is_made_of* which is a *material*. The *is_made_of* attribute is documented as "A reference to the material of which the layer is made" therefore it is easy to deduce that a *physical_layer* references a *material*.

3 Model integration requirements

There are a number of requirements which should be met by the integrated model. These requirements are set in order to maintain the consistency of the core model and to protect the existing users of the core model. The major requirement is to maintain **upward compatibility** between the core model and the integrated model. The upward compatibility constraint can be stated as "The integrated model must be a superset of the core model".

In other words, the integrated model must be able to convey all the information in the core model and users of the core model are guaranteed that they will not be affected by any changes made on the core model after the integration. In order to achieve this, the EDIF community has introduced these additional requirements on the integrated model:

- All objects which are defined in the core model must remain in the integrated model. That includes schemas, entities, attributes and constraints in an EXPRESS model.

- All objects which are defined in the core model must not have their names changed in the integrated model.

4 Model integration (Phase 1)

The model integration process is divided into two phases. The target of the first phase is to create a partially integrated model between the core EDIF model and the domain EXPRESS model. The first phase integration process is to determine how domain-specific objects are related to the objects already defined in the core EDIF model.

Before attempting any integration, it is necessary to have a reasonable understanding of the models that need to be integrated. The core EDIF model is already well understood, but the PCB EXPRESS model is less known. In the process of becoming familiar with the domain EXPRESS model, general mappings between the domain model and the core model inevitably spring to mind. However, these general mappings may not be the final mapping since they may be based on an incomplete understanding of the models. The high-level general mappings can be established by identifying categories of design information in the PCB model and determining how they are related to the design information already defined in the core EDIF model. Several categories of design information have been identified as common between the two models. Their use and possible merging are discussed below:

- instantiatable object information - in the core EDIF model, this provides template definitions whose occurrences can be used to support the implementation part of cell views. Similarly, in the PCB model, several classes of sub-layout definition are defined to support the description of a reusable bare board. However, most EDIF templates are designed specially for the schematic domain and they should not be overloaded to describe templates used primarily in the PCB domain. Hence, PCB-specific templates are not merged with any existing EDIF templates.

- connectivity information - in the core EDIF model, connectivity is divided into logical connectivity and structural connectivity. The logical connectivity defines the entire connectivity for a given level of hierarchy and the structural connectivity describes the implementation of the logical connectivity. In the PCB model, three levels of connectivity are identified; these are the functional netlist, the component netlist and the physical netlist. The functional netlist describes the complete connectivity that is to be realised by an assembled board and it corresponds to the logical connectivity in EDIF. Similarly, the component netlist and the physical netlist can be classified as the structural connectivity. However, structural connectivity in EDIF Version 3 0 0 is domain-specific. Hence, the PCB component netlist and physical netlist are not merged with existing EDIF structural connectivity objects which are defined specifically for other domains, e.g. schematic.

- geometry information - the concept of *geometric_shape* in the PCB model corresponds closely to the concept of *geometry_element* in the core EDIF model. Hence, it is possible to map PCB *geometric_shape* directly to EDIF *geometry_element*.

After some general high-level mappings are identified, the integration process continues by finding mappings between entities in the two models. The general procedures of the EDIF-PCB entity integration process are explained below.

4.1 Identifying top level entities

The owner of each entity in the PCB EXPRESS model is identified. The purpose of this activity is to determine the top level entities in the PCB EXPRESS model. Top level entities are most likely to be the core or major concepts. They represent the objects of interest which are good starting points from which the mapping should begin.

However, finding all top level entities is not a simple task. It involves the manual examination of all attribute relationships defined in the PCB EXPRESS model. It uses the additional information given on attribute relationships in the model translation process (see section 2).

4.2 Identifying areas of overlap

Areas of commonality and overlap between the core EDIF model and the PCB EXPRESS model are located. This is needed because every entity (i.e. object) in the domain EXPRESS model must be either created as a new entity or merged with an existing EDIF entity in the integrated model. The criterion used is to merge with existing entities whenever possible. Some possible ways of entity integration are described below.

If a domain entity corresponds to an entity defined in the core EDIF model and it conveys less information than the existing EDIF entity, an obvious solution is to replace the domain entity by the existing EDIF entity in the integrated model. However, domain experts should be consulted on whether or not they actually want the extra facilities provided by this existing EDIF entity.

For example, the PCB *identification_stamp* entity records the identification information of a PCB object. It includes the dates when the object was created and last updated. On the other hand, in the core EDIF model, the identification information of an object is captured by the *written* entity which records the time when the object was created or last modified. In addition, the *written* entity may also include program identification, human or organisation identification, or location information which can help to trace the origin of an EDIF object. It is obvious that the PCB *identification_stamp* entity can be mapped onto the EDIF *written* entity. PCB domain experts are consulted in order to decide whether or not it is appropriate to use the existing EDIF entity. In this case, it is decided that a separate PCB *identification_stamp* is not necessary for the integrated EDIF-PCB model and the existing *written* entity should be used instead.

In cases where a domain entity corresponds to an entity defined in the core EDIF model and it includes additional information which is not covered by the existing EDIF entity, changes to the existing EDIF entity will be required. EDIF experts should be consulted to agree on the changes made to the EDIF entity. For example, the EDIF *font* entity specifies information about the appearance of a text font and it is usually a reference to an externally defined standard text font. On the other hand, the PCB *text_font* can be either a reference to an externally defined standard text font (i.e. *standard_text_font*) or a set of rules for drawing individual text characters directly (i.e. *non_standard_text_font*). Hence, the core EDIF model should be extended to describe the character-based text font required by the PCB domain. One possible integration is to extend the EDIF *font* entity to include attributes belonging to a character-based text font. However, this leads to overloading of the existing *font* concept which is undesirable. Consequently, the current solution is to add a new *character_based_text_font* entity in the integrated model which describes the PCB *non_standard_text_font*. The existing EDIF *font* and the new *character_based_text_font* are related by sharing a common supertype. The common supertype *text_font* contains attributes which apply to both subtypes, e.g. font height and font width.

4.3 Determining place-holders for top level entities

If a domain entity describes an object which is not currently modelled in the core EDIF model, the simple solution is to add this domain entity to the integrated model. For example, the PCB *package* entity describes the mechanical data (information such as body volume, number of pins) of a physical package. It is a new concept for the core EDIF model, so the *package* entity is added to the integrated EDIF-PCB model.

Any top level domain entities that are not merged with concepts in the EDIF core model need to be placed somewhere in the resulting model. The two most likely places that such entities will be placed are in the *edif* entity or in the *library* entity.

The *edif* entity is the only top level entity in the core EDIF model and amongst other things contains libraries. The *library* entity is used to group related information. For example, the TTL 7400 series of components can be grouped into one or more EDIF libraries based on attributes such as functionality and performance.

The main criterion used in deciding whether a top level PCB domain model entity is placed in *edif* or in *library* is: does the entity represent information that it would be useful to transfer independently of a complete design? To answer this question it is necessary to consult with domain experts.

5 Model integration (Phase 2)

The first phase integration process produces the partially integrated model detailing the changes

required to the core EDIF model. The second phase of the integration process adds various EDIF features to the partially integrated model. The following are some of the important issues:

5.1 Adding internal/external distinction

In the core EDIF model, a *library* can be either internal or external to an EDIF file. An *external_library* represents a *library* which may be used within the file but is not explicitly defined with the file. That is, objects defined inside an *external_library* contain sufficient information to be used but they must not carry any detailed implementation information. On the other hand, objects defined inside an *internal_library* may contain the complete description. One of the reasons of making the internal/external distinction is that EDIF is designed for transferring design information between CAD systems. When design information is transferred across systems, it is desirable to transfer the same information only once. Hence, a transfer format needs a mechanism which supports references to previously transferred data. For example, a library of commonly used design information is transferred to the receiving system as an *internal_library*. In subsequent transfers, the sending system can refer to the previously transferred library at the receiving system as an *external_library*. This can lead to a considerable saving because duplicate transfer of design information is avoided.

The internal/external distinction in EDIF has made a considerable impact on the integration process. It is because in the PCB conceptual model, objects were defined without differentiating between interface information and implementation information. Thus, each attribute of a PCB entity needs to be examined carefully to determine whether it is part of the interface information or part of the implementation information. For example, in the PCB model, the *pin* entity has a *cross_section* attribute describing the cross-sectional shape of a pin, a *name* attribute describing the name of the pin and a *pin_length* attribute describing the length of the pin. Both the *cross_section* and the *pin_length* attributes are used to determine the physical shape of a pin; they are therefore classified as implementation information.

Figure 2 shows how the internal/external distinction is added to the *pin* entity. In the integrated

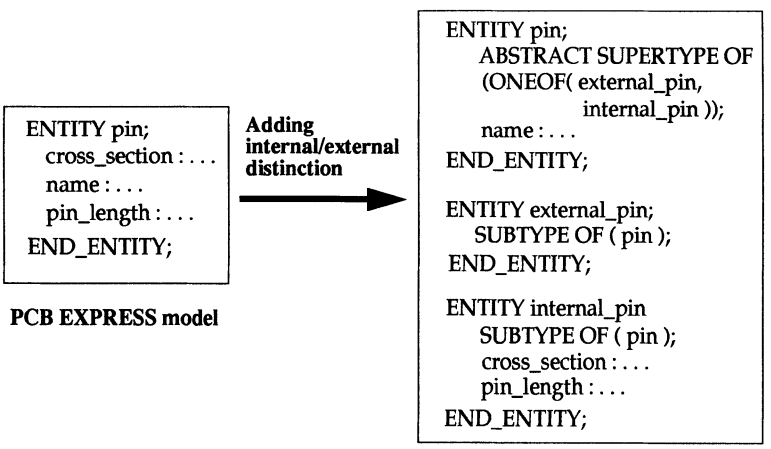

Figure 2: Adding internal/external distinction to PCB objects

EDIF-PCB model, a *pin* can be either an *internal_pin* or an *external_pin*. An *internal_pin* is defined within an *internal_library*; whereas an *external_pin* is defined within an *external_library*. The *name* attribute, which conveys the interface information, is placed in the *pin* entity. Both the *internal_pin* and the *external_pin* are subtypes of the *pin* entity and subsequently inherit the interface information. However, only *internal_pins* are allowed to contain implementation information, i.e. the *cross_section* and the *pin_length* attributes.

5.2 Adding units

In the PCB model, the units of measurement are not defined because it was felt that this level of detail was not required in a conceptual model. For example, the PCB model includes the concept of *distance* but does not specify whether this is measured in meters or inches. In EDIF, there is a *unit* mechanism which allows units to be defined in terms of SI units. Each measurement can then be associated with one of these user defined units. This mechanism is considered to be more than adequate by the PCB domain experts and so it is a simple task to merge any unit of measurement in the PCB domain model with the corresponding unit in the EDIF core model.

5.3 Adding name information and properties

EDIF has the concept of name information which is used to capture the object naming information in the CAD system. In addition, an EDIF object may have properties to convey CAD-specific information. PCB experts need to be consulted to determine which PCB objects have meaningful names or properties in CAD systems.

5.4 Maintaining consistency

When the EDIF-ised partially integrated model is added to the core EDIF model to create the final integrated EDIF-PCB model, it is important to maintain a consistent modelling style and naming mechanism in the fully integrated model. There are a number of issues which need to be considered:

- Names of entities and attributes should have a consistent style. For example, in the PCB model, the *package* entity has an attribute *has_pin_groups* which is a set of *pin_groups*. In the integrated model, the prefix of the attribute name "*has_*" is removed and *has_pin_groups* is renamed as *pin_groups*. It is felt that attribute prefixes like "*has_*" convey no additional information, so they are generally removed from the integrated model. This is consistent with the style of the existing EDIF Version 3 0 0 Information Model.

- There is a need to check that every entity added as a result of the integration process has an existence constraint defined either directly or indirectly. This is needed to ensure that no new entity is allowed to "float about" and the structure of EDIF is maintained.

- There is a need to check that the EDIF style of grouping mechanism is obeyed in any grouping constructs introduced in the integrated model. In EDIF, grouping is achieved by referencing the grouped objects. For example, an EDIF *signal_group* provides a grouping mechanism for *signals* and contains a list of references to other *signals* or *signal_groups*. However, in the PCB model, a *pin_group* "contains" a collection of *pins*. Consequently, PCB *pin_group* does not conform with EDIF style of grouping mechanism. Under these circumstances, the problem needs to be discussed between domain experts and EDIF experts so that they can decide whether or not a violation of the general mechanism should be permitted in the integrated model. In this case, it is agreed that the EDIF style of grouping mechanism should be used for modelling *pin_group*.

6 Problems encountered during model integration

During the integration process for PCB, new requirements have been added by the PCB domain experts. This has substantially increased the scope of the integration. These new areas included modelling of assembly drawing, technology rules and MCM (Multi Chip Modules). Hence, careful consideration must be given to ensure that the integrated model can be easily extended to cover these new areas.

Sometimes, it is difficult to decide which mapping should be chosen for a PCB object when there is more than one viable way to integrate it with EDIF. For example, in the PCB model, a *component* is an instantiation of a *part*. One possible mapping is to model *part* as a new class of instantiatable object in EDIF. Another possibility is to integrate it with the existing EDIF instantiation mechanism. The core EDIF model has the concept of instantiation: an *instance* is an instantiation of a *cluster* in a *cell*. Initially it seemed natural to adopt the second approach which maps the PCB *component* to the EDIF *instance* and the PCB *part* as a new type of view in a *cluster*. It was only much later in the integration process that it was decided this mapping is not appropriate because a PCB *part* is not a hierarchical object whereas an EDIF view contains instances of other cells to build a design hierarchy.

One of the model integration requirements listed in section 3 is that no object defined in the core EDIF model is allowed to have its name changed in the integrated model. This restriction also caused some discussion in the integration process. For example, the EDIF *font* entity currently describes an externally defined text font and it corresponds to the PCB *standard_text_font* entity. Ideally, the *font* entity should be renamed as *standard_text_font* because the new name describes exactly what it is. Also it is less confusing when the PCB *character_based_text_font* is introduced. However, name changing is considered to be an issue which affects other EDIF domains such as schematic. As a result, names of existing EDIF objects have not been changed in the current integrated model.

7 Conclusions

Having to merge quite distinct models in one go can be difficult. By dividing the integration process into two phases, the task of merging becomes manageable. Also, it is easier for both EDIF experts and domain experts to identify and review the changes to the core EDIF model.

Based on the experience derived from the EDIF-PCB integration, it can be concluded that the first phase of the model integration requires much more time and effort than the second phase. This is because during the first phase, various ways of mapping between objects need to be identified and the best possible integration has to be chosen. Sometimes, in order to evaluate trade-offs between various mappings, it is necessary to present several partially integrated models so that it is easier for the reviewers to gain an overall picture of different ways of integration.

The integration of a domain conceptual model with the EDIF Information Model requires a good understanding of both models. This understanding cannot be derived from the EXPRESS models alone since much of the necessary information is not contained in an EXPRESS model but is to be found in accompanying documentation. For example, a *part* in the conceptual model is identified as an instantiatable object because the description of *component* in the accompanying documentation states that a *component* is an instantiation of a *part*. This information cannot be directly derived from the attributes of *part* and *component* or any relationship between them. The fact that much of the information needed to merge models is not in a formal form that can be readily processed by a computer means that complete automation of the model merging process is not possible. More experience in merging models manually needs to be obtained to identify common areas in which there is potential for automation.

Finally, it is important to note that the integration procedures outlined in this paper should be taken as guidelines only. Since it is not possible to predict what format or form any other domain conceptual models (e.g. Test domain, VHDL, CFI) will take, it is difficult at this stage to come up with a general principle which can be applied in all possible cases.

8 Bibliography

1. Electronic Industries Association. *Electronic Design Interchange Format Version 3 0 0*, EIA-618, Volume 1, Dec 1993.

2. Electronic Industries Association. *Electronic Design Interchange Format Version 3 0 0*, EIA-618, Volume 2, Dec 1993.

3. EDIF PCB Technical Sub-Committee. *Conceptual Model of a PCB Version 12*, Nov 1993.

4. International Standard Organization STEP ISO/TC 184/SC4/WG5. *EXPRESS Language Reference Manual*, N14 edition, April 1991.