

Translating Descriptions of a Viewpoint among Different Representations

Nigel Stanger

**The Information Science
Discussion Paper Series**

Number 2000/11
May 2000
ISSN 1172-6024

University of Otago

Department of Information Science

The Department of Information Science is one of six departments that make up the School of Business at the University of Otago. The department offers courses of study leading to a major in Information Science within the BCom, BA and BSc degrees. In addition to undergraduate teaching, the department is also strongly involved in post-graduate research programmes leading to MCom, MA, MSc and PhD degrees. Research projects in spatial information processing, connectionist-based information systems, software engineering and software development, information engineering and database, software metrics, distributed information systems, multimedia information systems and information systems security are particularly well supported.

The views expressed in this paper are not necessarily those of the department as a whole. The accuracy of the information presented in this paper is the sole responsibility of the authors.

Copyright

Copyright remains with the authors. Permission to copy for research or teaching purposes is granted on the condition that the authors and the Series are given due acknowledgment. Reproduction in any form for purposes other than research or teaching is forbidden unless prior written permission has been obtained from the authors.

Correspondence

This paper represents work to date and may not necessarily form the basis for the authors' final conclusions relating to this topic. It is likely, however, that the paper will appear in some form in a journal or in conference proceedings in the near future. The authors would be pleased to receive correspondence in connection with any of the issues raised in this paper, or for subsequent publication details. Please write directly to the authors at the address provided below. (Details of final journal/conference publication venues for these papers are also provided on the Department's publications web pages: <http://www.otago.ac.nz/informationscience/pubs/publications.htm>). Any other correspondence concerning the Series should be sent to the DPS Coordinator.

Department of Information Science
University of Otago
P O Box 56
Dunedin
NEW ZEALAND

Fax: +64 3 479 8311

email: dps@infoscience.otago.ac.nz

www: <http://www.otago.ac.nz/informationscience/>

Translating Descriptions of a Viewpoint among Different Representations

Nigel Stanger

University of Otago, Dunedin, New Zealand

nstanger@infoscience.otago.ac.nz

Abstract

An important part of the systems development process is building models of real-world phenomena. These phenomena are described by many different kinds of information, and this diversity has resulted in a wide variety of modelling representations. Some types of information are better expressed by some representations than others, so it is sensible to use multiple representations to describe a real-world phenomenon. The author has developed an approach to facilitating the use of multiple representations within a single viewpoint by translating descriptions of the viewpoint among different representations. An important issue with such translations is their quality, or how well they map constructs of one representation to constructs of another representation. Two possible methods for improving translation quality, heuristics and enrichment, are proposed in this paper, and a preliminary metric for measuring relative translation quality is described.

1 Introduction

To more completely model a phenomenon, Finkelstein et al. [1] suggested using multiple descriptions of the phenomenon expressed using different representations. For example, a data flow diagram (DFD) is a description of constituent business processes expressed using a representation suited to showing the flow of information among these processes; whereas an entity-relationship diagram (ERD) describes clearly the structure and associations among data elements. The benefits from using different descriptions of the same phenomenon are recognised in current modelling approaches such as the Unified Modeling Language (UML) [2], in which real world phenomena are described using a combination of use cases, class hierarchy diagrams, sequence diagrams, state charts, and so on.

The author has developed an approach to facilitating the use of multiple representations within a single viewpoint by translating descriptions of the viewpoint among

different representations [3]. These translations are defined by a collection of *rules* that specify how to map constructs of a source representation to constructs of a target representation. A key issue with such translations is how well they map constructs of the source representation onto constructs of the target representation, referred to here as the *quality* of a translation [4].

In this paper are proposed two possible methods for improving the quality of translations between representations: use of *heuristics* which allow additional information to be generated automatically by a translation; and *enrichment*, in which the translation elicits missing information from the developer. Also described is a preliminary metric that has been used to show that heuristics can improve the quality of translations.

2 Terminology and notation used in this paper

Finkelstein et al. [1] suggested using multiple representations in the context of a viewpoint-oriented approach to systems analysis and design, which acknowledges that there may often be multiple, possibly conflicting interpretations or *perspectives* of the phenomenon being modelled. For example, the Chief Executive, marketing manager, sales personnel and accountant are likely to have different perspectives, each having correspondingly different descriptions of the business. A *viewpoint* is the formalisation of a particular perspective [5], as shown at the top of Figure 1.

Viewpoints provide a useful context within which to discuss translations among multiple representations, so the author [3, 6, 7] has developed the viewpoint-based framework shown in Figure 1. This framework was synthesised from earlier work by Finkelstein et al. [1], Easterbrook [5] and Darke & Shanks [8].

As shown in Figure 1, a *viewpoint* comprises one or more *descriptions*, each expressed using a suitable *representation*. A representation comprises two parts: a tech-

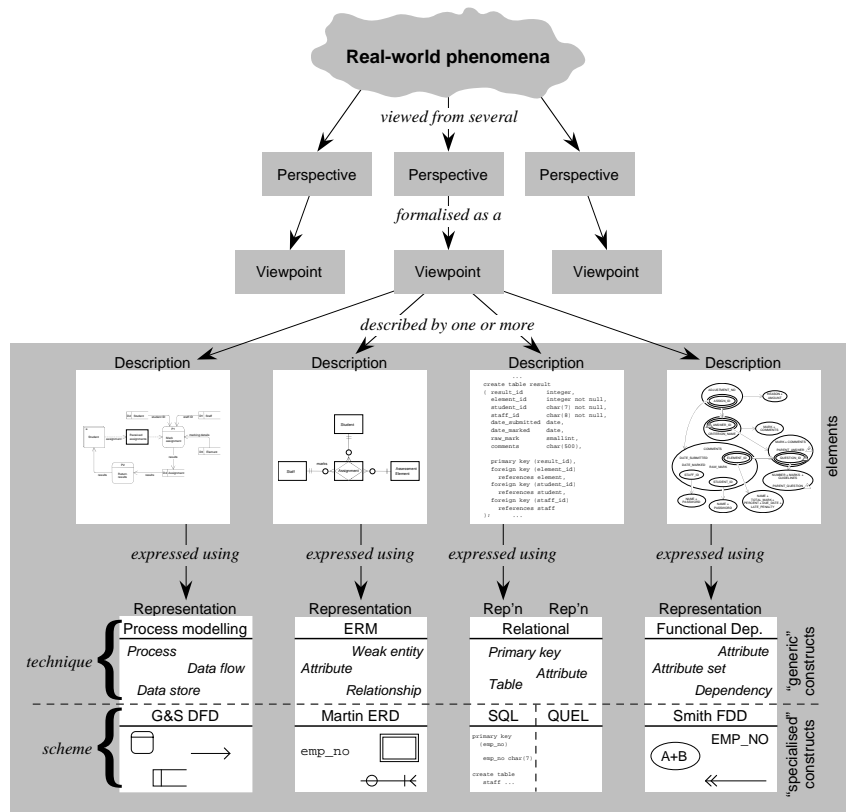


Figure 1: The viewpoint/representation/description framework

nique and a scheme. A representation's *technique* specifies the generic *constructs* of the representation, such as classes, relations and attributes. A representation's *scheme* comprises constructs specialised from the generic constructs of the technique, and includes the 'symbols' required to express these constructs; for example, boxes for UML classes, SQL `create table` statements for relations and so on. While a given technique may have several possible schemes, the combination of a particular technique and scheme forms a single representation. For example, SQL/92 and QUEL are two possible schemes that may be used in association with the relational technique, but the combinations (*Relational, SQL/92*) and (*Relational, QUEL*) form two distinct representations, as illustrated in Figure 1.

Representations must be instantiated in some way in order to describe a viewpoint. The instantiation of a representation can be thought of as a collection of 'statements' that describe the viewpoint, and is thus referred here as a *description* (the terms 'model' or 'schema' could perhaps be used, but these terms are somewhat ambiguous). The 'statements' of a description are expressed by *elements*,

which can be thought of as instantiations of a representation's constructs; for example, the **Customer** actor in a use case diagram for a sales system (an instantiation of the **ACTOR** construct), or the **Product** table in an SQL/92 schema definition (an instantiation of the **TABLE** construct).

In practical terms, a representation can be thought of as the combination of a generic modelling approach (the technique) with a specific version of that approach (the scheme, which includes the notation used). The representation is then used to build descriptions of a particular viewpoint.

2.1 Abstract notation

The author has developed an abstract notation for writing representation, description, construct and element expressions [7]. The notation is derived from the data transfer notation of Pascoe and Penny [9], and is summarised in Table 1. (There are additional notations for specifying translations which are not used in this paper and have thus been omitted.)

The notation $D_i(V, T, S)$ denotes that description D_i of viewpoint V is expressed using constructs of technique

Table 1: Summary of representation, description, construct and element notation

Notation	Associated term	Definition
V	Viewpoint	A formalisation of a particular perspective on a real-world phenomenon.
T	Technique	A collection of generic constructs that form a modelling ‘method’, for example, the relational model or entity-relationship approach.
S	Scheme	A collection of specialised constructs that form a specific modelling ‘notation’ within a technique, for example, SQL/92 or Martin ERD notation.
$\mathfrak{R}(T, S)$ or \mathfrak{R}	Representation	Representation \mathfrak{R} comprises constructs defined by the combination of technique T and scheme S .
$D_i(V, T, S)$ or D	Description	Description D_i of viewpoint V is expressed using constructs of technique T and scheme S .
$\mathfrak{R}(T, S)$ [CON], \mathfrak{R} [CON], or CON	} Construct of a representation	CON specifies a construct of representation $\mathfrak{R}(T, S)$.
$D_i(V, T, S)$ [e : CON], D_i [e : CON], or D_i [e]	} Element of a description ...	e specifies an element (instantiated from construct CON) of description $D_i(V, T, S)$.

T and scheme S (this may be abbreviated to D_i when V , T and S are clear). Thus, $D_1(V_p, E-R, ERD_{Martin})$ denotes a description D_1 of the viewpoint V_p that is expressed using constructs of the entity-relationship technique ($E-R$) and the Martin ERD scheme (ERD_{Martin}) [10].

The notation $\mathfrak{R}(T, S)$ denotes a representation \mathfrak{R} that comprises a collection of constructs defined by the combination of technique T and scheme S (this may be abbreviated to \mathfrak{R} when T and S are clear). Thus, $\mathfrak{R}_e(E-R, ERD_{Martin})$ denotes the representation \mathfrak{R}_e formed by combining the constructs of the entity-relationship technique ($E-R$) with the Martin ERD scheme (ERD_{Martin}).

Constructs are the fundamental components of a representation, whereas elements are the fundamental components of a description. Given a representation $\mathfrak{R}(T, S)$, a construct CON of \mathfrak{R} is denoted by $\mathfrak{R}(T, S)$ [CON], or, if T and S are clear, simply \mathfrak{R} [CON]. Often \mathfrak{R} may also be clear from the context, allowing the \mathfrak{R} [] notation to also be omitted, leaving just CON. The name of the construct is denoted by SMALL CAPS. For example, the regular entity construct of the representation $\mathfrak{R}_e(E-R, ERD_{Martin})$ is denoted by $\mathfrak{R}_e(E-R, ERD_{Martin})$ [REGULARENTITY] or \mathfrak{R}_e [REGULARENTITY].

The construct CON can be thought of as analogous to the concept of a relational domain in that it specifies a pool of possible ‘values’ from which an element e may be drawn. The notation e : CON is used to denote that e is a member of the set of all possible elements corresponding to the construct CON.

Now consider a description $D_i(V, T, S)$. An element e of D_i (instantiated from construct $\mathfrak{R}(T, S)$ [CON]) is denoted by $D_i(V, T, S)$ [e : CON], or, if V , T and S are clear, simply D_i [e : CON]. The construct may also be omitted if it is clear from the context, that is, D_i [e]. For example, the Customer regular entity of the description $D_1(V, E-R, ERD_{Martin})$ is denoted by $D_1(V, E-R, ERD_{Martin})$ [*customer* : REGULARENTITY] or D_1 [*customer*].

3 Translating descriptions among representations

A representation comprises a collection of constructs that are instantiated to form the elements of descriptions. The translation of a description from a source representation \mathfrak{R}_s to a target representation \mathfrak{R}_t can therefore be decom-

posed into a collection of translations of the elements that make up the description. The translation of one or more elements of a source description to one or more elements of a target description is defined by a *rule* that specifies a mapping from a collection of constructs of \mathfrak{R}_s to some collection of constructs of \mathfrak{R}_t [11, 12]. This mapping may have constraints that specify pre- and post-conditions for elements translated by the rule. Pre-conditions ensure that a rule is applied only to appropriate source elements. Post-conditions enforce the semantics of the target representation by ensuring that elements generated by a rule make sense in the context of the target representation. Both pre- and post-conditions are referred to here as *invariants* [13], as a pre-condition when a rule is applied in one direction ('left-to-right') can become a post-condition when the rule is applied in the opposite direction ('right-to-left').

The application of a rule should always result in a target structure that is semantically consistent with the source description, but in general it is unlikely that a set of rules can be defined to map completely from \mathfrak{R}_s to \mathfrak{R}_t . This is because some constructs of \mathfrak{R}_s may not be directly mappable onto constructs of \mathfrak{R}_t , or vice-versa. This mismatch can sometimes be ameliorated by use of *heuristics*, which are similar to rules except that the application of a heuristic

will *generally* result in a target structure that is semantically consistent with the source, but not always. Heuristics can be thought of as 'rules of thumb' that usually but not always produce the correct result.

Rules generally only translate the structure or syntax of a description. Heuristics also translate structure, but can also affect the semantics of the encompassing viewpoint by making explicit semantics that may be implicit in the original description. Consider the functional dependency description $D_1(V_{staff}, FuncDep, FDD_{Smith})$ expressed using Smith's notation [14, 15] and shown in Figure 2(a). This description has a collection of domain flag elements that all reference the same attribute element (IRD_NUMBER). Now suppose there is a rule specifying that a DOMAINFLAG construct maps to a RELATIONSHIP construct in an ERD. The description $D_2(V_{staff}, E-R, ERD_{Martin})$ that results from applying this rule is shown in Figure 2(b). Examining D_2 , it is apparent that the **Salary_staff**, **Wage_staff** and **Salesrep** entities are actually subtypes of the **Staff** entity. A rule cannot be defined to this effect, however, because a domain flag does not always imply a type hierarchy.

Instead, a heuristic could be defined that specifies a mapping from a collection of DOMAINFLAG constructs to a TYPEHIERARCHY construct. The likelihood of this be-

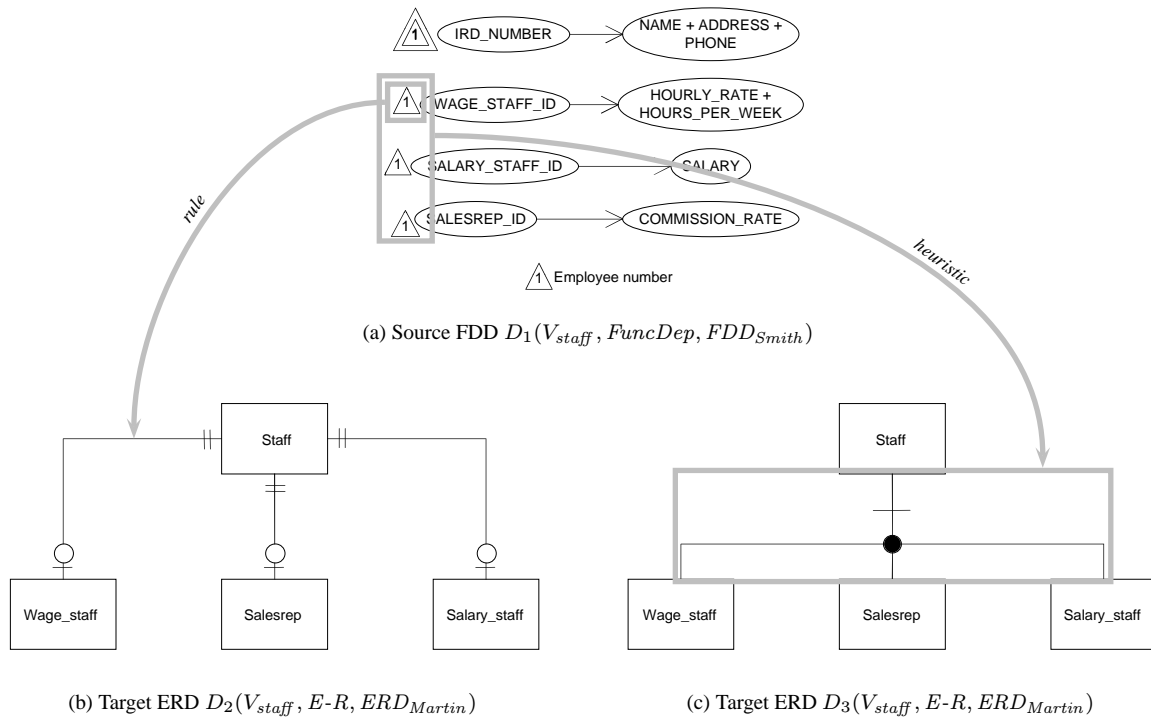


Figure 2: Example of applying a heuristic

ing the correct interpretation increases as the number of domain flags referencing the attribute increases. A reasonable assumption is that if three or more domain flags reference the same attribute, they should be mapped to a type hierarchy (this would be defined as an invariant on the heuristic). The result of applying such a heuristic is shown in Figure 2(c). Compare D_2 and D_3 — without the heuristic, the type hierarchy implicit in D_1 and D_2 would not be drawn out in D_3 . This heuristic will of course not always apply, but will usually produce the correct result.

In effect, the application of heuristics can cause an increase in the explicit semantic content of a viewpoint, by drawing out semantics that are implicit in the source description. Thus, in the example shown in Figure 2, the explicit semantic content of the viewpoint is increased by making explicit the type hierarchy implicit in the source FDD.

In summary, the translation of a description from one representation to another can be decomposed into a collection of translations of elements. These element translations are defined by a collection of rules and heuristics, which specify mappings between collections of constructs in the source and target representations. Rules always produce semantically consistent results, whereas heuristics can sometimes produce semantically inconsistent results. A simple prototype system called *Swift* has been implemented that demonstrates use of rules and heuristics in description translations [16, 17].

3.1 Translation quality

Every representation has boundaries to what may be expressed using the constructs of that representation. This

is referred to here as the *expressive power* of a representation, and covers the constructs of a representation and any semantic constraints that may exist between constructs. Representations may overlap in terms of expressive power, such that some subset of the constructs of one representation can be mapped to constructs of another representation. This concept is referred to here as the *expressive overlap* between two representations.

The nature of the expressive overlap between a particular pair of representations $\mathfrak{R}_p(T_i, S_j)$ and $\mathfrak{R}_q(T_k, S_l)$ can fall into one of four categories, as illustrated in Figure 3. The four categories of expressive overlap are:

Disjoint There is no expressive overlap between \mathfrak{R}_p and \mathfrak{R}_q , as shown in Figure 3(a), and translating descriptions between them is therefore typically impossible.

Intersecting There is a partial expressive overlap between \mathfrak{R}_p and \mathfrak{R}_q , as shown in Figure 3(b), and it is therefore possible to partially translate descriptions from \mathfrak{R}_p to \mathfrak{R}_q , and vice versa.

Inclusive One representation is contained entirely within the other, for example, \mathfrak{R}_p contains \mathfrak{R}_q in Figure 3(c). Anything that can be expressed using constructs of \mathfrak{R}_q may also be expressed using constructs of \mathfrak{R}_p , but not vice versa. It is therefore possible to translate descriptions completely from \mathfrak{R}_q to \mathfrak{R}_p , but only partially from \mathfrak{R}_p to \mathfrak{R}_q .

Equivalent Anything that can be expressed using constructs of \mathfrak{R}_p can also be expressed using constructs of \mathfrak{R}_q , and vice versa, as shown in Figure 3(d). It is therefore possible to translate descriptions completely in either direction.

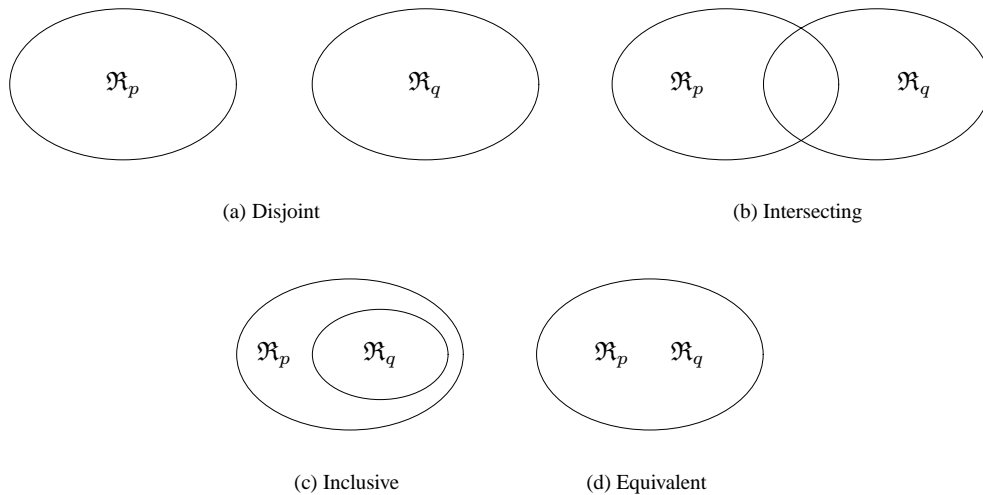


Figure 3: Four categories of expressive overlap

The *quality* of a translation is defined as how completely it maps constructs from the source representation to the target representation [4]. The expressive overlap of a pair of representations determines the maximum amount of information that may be translated from one representation to the other, and hence the maximum quality of any translation between those representations. The intersecting and inclusive categories are of the most interest, because translating a description from a more expressive representation to a less expressive one can result in a ‘loss’ of information, whereas the reverse translation can result in the need to ‘gain’ information during the translation in order to build a sensible target description.

4 Improving translation quality

In an environment that facilitates the use of multiple representations, information loss may not be as great a concern as it is in more typical data translation problems. This is because the information that is ‘lost’ during a description translation is still held in the source description. All descriptions will presumably be held in the same repository, so it may be possible to ‘restore’ information that is ‘lost’ during a translation when the translation is applied in the reverse direction. For instance, if the data flow description $D_1(V, DataFlow, DFD_{G\&S})$ is translated into the functional dependency description $D_2(V, FuncDep, FDD_{Smith})$, the names of data stores may not be translated because of differences in the expressive powers of the two representations. The data store names still exist in D_1 , however, so if D_2 is at some later stage used to incrementally update D_1 , or translated to a new DFD D_3 , then it is in principle possible to extract the data store names from the original D_1 . This would require some means of tracking the translation ‘history’ of descriptions.

A less obvious concern that arises when performing description translations is that it may be necessary to ‘gain’ information that does not already exist when performing a translation. For example, when translating from a functional dependency description to an entity-relationship description, the names of the entities that are created must be generated or acquired in some way. This sort of situation would most commonly occur with inclusive representations when translating a description from the ‘contained’ representation to the ‘containing’ representation, but could also occur with intersecting representations.

Information ‘gain’ is a particular issue with translations among representations with different techniques, as the expressive powers of two representations with different techniques are more likely to be divergent than the expressive powers of two representations that share the same tech-

nique. The effect will be less severe in translations that change only the scheme.

The problem is determining what information can and cannot be generated automatically during a translation, and why. Information that cannot be automatically generated and that is essential to build a syntactically correct target description (such as the names of tables in SQL) must be acquired somehow, otherwise it will not be possible to complete the translation. One way to improve the quality of a translation would therefore be to acquire information from the developer in some way. This process of acquiring information is termed *enrichment*, and may occur before, during and after a translation.

Enrichment performed before a translation (‘pre-enrichment’) involves pre-populating the viewpoint with information that will be useful to the target description, and is analogous to Su & Fang’s [18] notion of modifying and extending the semantics of a schema before translating it. For example, before translating from an FDD to an ERD, template entities with appropriate names but no attributes could be created by the developer, to filled in later by the translation.

Enrichment performed during a translation requires the developer to at least supply any information that is essential to properly build the target description, and that has not already been supplied. Using the same example as above, if entity names were not defined prior to the translation, then the developer would be requested to enter suitable names for each entity as it was generated by the translation. Non-essential information could also be included at this stage.

Enrichment after a translation (‘post-enrichment’) involves adding any remaining missing information that was not automatically generated during the translation or entered prior to the translation. In effect, the developer is refining the resultant description.

Another way to improve the quality of a translation is the application of heuristics. Heuristics allow the *automatic* translation of more information than would normally be possible using rules, by drawing out semantics that are implicit in the source description, as noted in the previous section. Adding heuristics can reduce the amount of enrichment that is required, and produces an improvement in translation quality, as less information is lost or needs to be gained during the translation.

An important issue with use of heuristics is that they may sometimes produce semantically inconsistent results, which could result in an invalid viewpoint. This issue will typically arise when heuristics are applied without regard to the context in which they are being applied. That is, semantically inconsistent results will be produced in cases where the heuristic was not appropriate and should not have been applied. The applicability of a particular heuristic, however, will typically be difficult (if not impossible) to

determine in an automatic manner. Fortunately, the enrichment process provides a practical solution to this issue. Rather than applying heuristics regardless of context, the developer can be notified that a heuristic is about to be applied and what the result of applying the heuristic will be. The developer can then use their contextual and semantic knowledge of the viewpoint to determine whether this is a correct result, and thus choose whether or not the heuristic should be applied.

Heuristics can increase the explicit semantic content of a viewpoint, so ensuring that heuristics are manually activated provides the benefit of making the developer aware of the semantic changes that are occurring during the translation. This process may also help identify possibilities that the developer may not otherwise have considered, thus promoting a more complete understanding by the developer of the phenomenon being modelled.

Both heuristics and enrichment have been implemented in the Swift prototype mentioned earlier, and have been found informally to produce improvements in translation quality. It would be useful, however, to actually measure such quality gains. In the next section is described a preliminary metric for measuring relative translation quality, which has been used to show that heuristics can produce improvements in translation quality [3].

5 Measuring relative translation quality

In order to determine whether use of heuristics has an impact on translation quality, the author is developing metrics for measuring translation quality. An overview of a preliminary metric is presented here.

Hull's concept of *relative information capacity* [19] provides a basis for comparing the information content of schemas in the context of schema integration, and Miller's *schema intension graphs* [20] are a tool that may be used to determine the relationship between the relative information capacities of two schemas. Representation definitions can be interpreted as a form of schema, so relative information capacity may also be applied to representations. In particular, the author has found that the expressive power of a representation can be characterised by the information capacity of its definition. Consequently, schema intension graphs may be used to determine the category of expressive overlap between two representations, providing a relative measure of translation quality.

Schema intension graphs provide an alternate graphical representation for describing schemas, and comprise:

- a collection of nodes representing typed domains (that is, sets of data values);

- a collection of labelled edges representing binary relations between domains; and
- annotations on the edges that represent simple constraints on the binary relations.

In Figure 4 is shown an example of a schema intension graph (SIG) that corresponds to an entity-relationship diagram of a used cars business.

It is possible to determine the relationship between the relative information capacities of two schemas by comparing SIGs for the two schemas. Intuitively, if two SIGs are isomorphic, then the corresponding schemas are equivalent [20]. An isomorphism occurs between two SIGs when the graphs are structurally identical and corresponding nodes in the two SIGs have compatible semantics. Isomorphisms may also be achieved by performing various equivalence preserving and information capacity augmenting transformations on one of the SIGs [21].

The information capacity of a schema determines the set of all valid instances of that schema within the context of the phenomenon that it models [22]. A schema defines a set of valid database structures for a given phenomenon. Combined with the set of all data relating to the phenomenon, the schema may be used to generate all possible database states for the phenomenon. By extension, the information capacity of a representation definition determines the set of all valid descriptions that may be expressed using the representation; that is, it determines the extent of what may be expressed using that representation. This is effectively identical to the notion of expressive power introduced in Section 3.1. That is, the expressive power of a representation can be characterised by the information capacity of the representation's definition. This means that SIGs may be used to determine any relationship between the expressive powers of two representations, and thus the nature of the expressive overlap between those representations, which in turn gives an indication of the maximum quality of any translations between those representations.

Given two representations $\mathfrak{R}_p(T_1, S_1)$ and $\mathfrak{R}_q(T_2, S_2)$, the expressive overlap of the pair can be categorised by building SIGs corresponding to the definitions of the two representations, and comparing them as follows:

- If the two SIGs are isomorphic, or one SIG can be transformed by a series of equivalence preserving transformations to be isomorphic to the other, then the expressive powers of \mathfrak{R}_p and \mathfrak{R}_q are equivalent.
- If the two SIGs are not isomorphic, but the SIG for \mathfrak{R}_p can be transformed by a series of information capacity augmenting transformations to be isomorphic to the SIG for \mathfrak{R}_q , then the expressive power of \mathfrak{R}_q is inclusive of that of \mathfrak{R}_p (or vice versa).

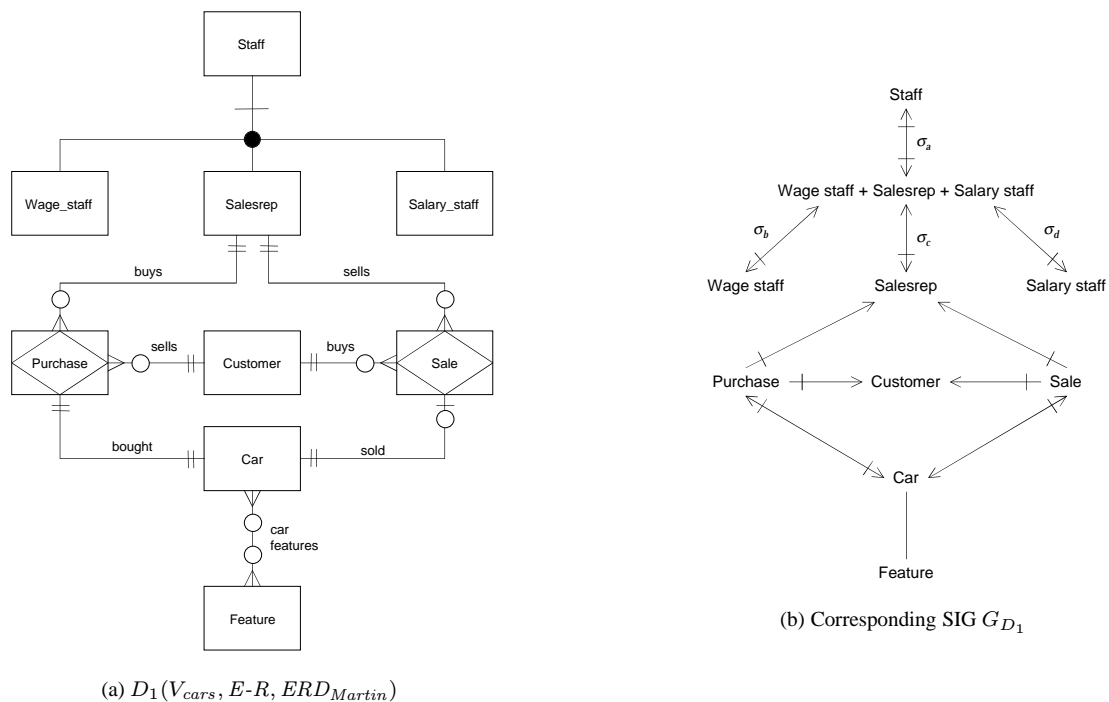


Figure 4: Example of a schema intension graph

- If the two SIGs are not isomorphic, but one or more subgraph(s) of the SIGs are either isomorphic, or can be transformed to be isomorphic, then \mathfrak{R}_p and \mathfrak{R}_q have an intersecting expressive overlap.
- If the two SIGs are not isomorphic, and no subgraphs of the SIGs are isomorphic and cannot be transformed to be isomorphic, then \mathfrak{R}_p and \mathfrak{R}_q have a disjoint expressive overlap.

The author has used these concepts to develop a preliminary metric that provides a relative measure of translation quality [3]. SIGs are built for both the source and target representations, and the extent of overlap is calculated by tagging nodes and edges in each SIG corresponding to constructs that are mapped by the translation. These then counted to provide an indication of the relative quality of the translation. An example of this approach is shown in Figure 5 for the translation from $\mathfrak{R}_f(FuncDep, FDD_{Smith})$ to $\mathfrak{R}_e(E-R, ERD_{Martin})$ (tagged nodes and edges are shown in **bold italic**).

This metric has been successfully used by the author to show that use of heuristics has a positive impact on the quality of translations, but unfortunately the metric is only useful for comparing the *relative* quality of two or more translations. It does not provide a quantitative measure of translation quality, so further research is required.

6 Summary

There are many diverse types of information to be modelled when building an information system, so it is sensible to use multiple representations to describe real-world phenomena. The author has developed an approach to facilitating the use of multiple representations within a single viewpoint by translating descriptions of the viewpoint among different representations. An important issue with such translations is their *quality*, or how well they map constructs of one representation to constructs of another representation. In this paper were proposed two methods of improving translation quality, *heuristics* and *enrichment*.

Heuristics improve translation quality by automatically translating information that could not normally be translated. In particular, heuristics can draw out implied semantics from the source description, and make them explicit in the target description. Heuristics may, however, sometimes produce incorrect results.

Enrichment improves translation quality by eliciting additional information from the developer that cannot be generated automatically. Enrichment may occur before, during and after a translation.

Also described was a preliminary metric for comparing the relative quality of translations. Relative information capacity can be used to characterise the expressive power of

- Science, University of Otago, Dunedin, New Zealand, Dec. 1999.
- [4] R. Pascoe and J. Penny, "Construction of interfaces for the exchange of geographic data," *International Journal of Geographical Information Systems*, vol. 4, no. 2, pp. 147–156, 1990.
- [5] S. M. Easterbrook, *Elicitation of Requirements from Multiple Perspectives*. PhD thesis, Imperial College of Science Technology and Medicine, University of London, London, 1991.
- [6] N. Stanger and R. Pascoe, "Environments for viewpoint representations," in *Proceedings of the Fifth European Conference on Information Systems (ECIS'97)* (R. Galliers, S. Carlsson, C. Loebbecke, C. Murphy, H. Hansen, and R. O'Callaghan, eds.), vol. I, (Cork, Ireland), pp. 367–382, Cork Publishing, June 19–21 1997.
- [7] N. Stanger, "A viewpoint-based framework for discussing the use of multiple modelling representations," Discussion Paper 2000/09, Department of Information Science, University of Otago, Dunedin, New Zealand, Apr. 2000.
- [8] P. Darke and G. Shanks, "Stakeholder viewpoints in requirements definition: A framework for understanding viewpoint development approaches," *Requirements Engineering*, vol. 1, pp. 88–105, 1996.
- [9] R. T. Pascoe and J. P. Penny, "Constructing interfaces between (and within) geographical information systems," *International Journal of Geographical Information Systems*, vol. 9, no. 3, pp. 275–291, 1995.
- [10] J. Martin, *Information Engineering, Book II: Planning and Analysis*. Englewood Cliffs, New Jersey: Prentice-Hall, revised ed., 1990.
- [11] P. Atzeni and R. Torlone, "Schema translation between heterogeneous data models in a lattice framework," in *Database Applications Semantics, Proceedings of the Sixth IFIP TC-2 Working Conference on Data Semantics (DS-6)* (R. Meersman and L. Mark, eds.), (Stone Mountain, Atlanta, Georgia), pp. 345–361, Chapman & Hall, May 30–June 2 1995.
- [12] P. Atzeni and R. Torlone, "Management of multiple models in an extensible database design tool," in *Proceedings of the Fifth International Conference on Extending Database Technology (EDBT'96)* (P. Apers, M. Bouzeghoub, and G. Gardarin, eds.), vol. 1057 of *Lecture Notes in Computer Science*, (Avignon, France), pp. 79–95, Springer-Verlag, Mar. 25–29 1996.
- [13] R. W. Amor, *A Generalised Framework for the Design and Construction of Integrated Design Systems*. PhD thesis, Department of Computer Science, University of Auckland, Auckland, New Zealand, May 1997.
- [14] H. C. Smith, "Database design: Composing fully normalized tables from a rigorous dependency diagram," *Communications of the ACM*, vol. 28, no. 8, pp. 826–838, 1985.
- [15] N. Stanger, "Modifications to Smith's method for deriving normalised relations from a functional dependency diagram," Discussion Paper 99/23, Department of Information Science, University of Otago, Dunedin, New Zealand, Dec. 1999.
- [16] N. Stanger and R. Pascoe, "Exploiting the advantages of object-oriented programming in the implementation of a database design environment," in *Proceedings of the Joint 1997 Asia Pacific Software Engineering Conference and International Computer Science Conference (APSEC'97/ICSC'97)*, (Hong Kong), IEEE Press, Dec. 2–5 1997.
- [17] N. Stanger and R. Pascoe, "Exploiting the advantages of object-oriented programming in the implementation of a database design environment," Discussion Paper 97/08, Department of Information Science, University of Otago, Dunedin, New Zealand, 1997.
- [18] S. Su and S. Fang, "A neutral semantic representation for data model and schema translation," Technical report TR-93-023, University of Florida, Gainesville, Florida, July 1993.
- [19] R. Hull, "Relative information capacity of simple relational database schemata," *SIAM Journal on Computing*, vol. 15, pp. 856–886, Aug. 1986.
- [20] R. Miller, Y. Ioannidis, and R. Ramakrishnan, "Schema intension graphs: A formal model for the study of schema equivalence," Technical report CS-TR-94-1185, Department of Computer Sciences, University of Wisconsin-Madison, Madison, Wisconsin, Jan. 1994.
- [21] R. Miller, Y. Ioannidis, and R. Ramakrishnan, "Schema equivalence in heterogeneous systems: Bridging theory and practice," *Information Systems*, vol. 19, no. 1, pp. 3–31, 1994.
- [22] R. Miller, *Managing Structural Heterogeneity*. PhD thesis, Department of Computer Sciences, University of Wisconsin-Madison, Madison, Wisconsin, 1994.