Theories of Meaning in Schema Matching: An Exploratory Study

Joerg Evermann

Faculty of Business Administration
Memorial University of Newfoundland
St. John's, NL, Canada

Abstract

Schema-matching is an important step in database integration. It identifies elements in two or more databases that have the same meaning. A multitude of schema matching methods have been proposed, but little is known about how humans assign meaning to database elements or assess the similarity of meaning of database elements. This paper presents an initial experimental study based on five theories of meaning that compares the effects of seven factors on the perceived similarity of database elements. Implications for schema matching research are discussed and guidance for future research is offered.

Key words: database, integration, schema matching, meaning, experiments

1 Introduction

Database integration is an increasingly important activity to ensure the continuing performance and competitive advantage of businesses. Information system development often requires integrating existing legacy systems and their databases. Business intelligence through data warehousing requires the integration of data from multiple transaction processing systems for decision support. Electronic business between organizations requires the integration of trade partners' business data for efficient inter-organizational business processes.

Database integration is a process with multiple steps, leading from the identification of the databases to the testing of the integrated system. The central

Email address: jevermann@mun.ca (Joerg Evermann).

step of the database integration process is the identification of those elements in the schemata of the database that match each other. This step is termed *schema matching*.

Many different automatic or semi-automatic methods for matching database schema elements have been proposed [1,2]. These are useful in the absence of the original database designers, the lack of detailed documentation, or the presence of very large databases. In these cases, all the information available is that in the database itself, primarily its schema and instances. The fact that all method performance evaluations show often significantly less than 100% success at matching [3–12], combined with the ongoing research effort in the field, shows that the problem is not yet completely solved.

Consider a situation that requires the integration of production management data with marketing data for a decision support system. The marketing database contains information about parts and articles, while the production database contains information about products and components (Fig. 1). How does a database integrator decide which of these elements match each other? The main thesis of this paper is that the database integrator matches elements if they have the same meaning. In the example in Fig. 1, the schema elements 'Product' and 'Article' are matched if the database integrator decides they have the same meaning. Similarly, the elements 'ProductID' and 'SerialNum' are matched if they have the same meaning to the database integrator [13].

In the example in Fig. 1, how does the database integrator decide whether 'Product' and 'Article' have the same meaning? What criteria does she apply? Is it because products and articles refer to the same things (e.g. things on the factory floor)? Is it because products and components are in the same kind of relationship as articles and parts? Or is it perhaps because products and articles are described by similar features, i.e. the primary keys are both of type character? Or perhaps it is because the data in the corresponding tables 'Product' and 'Article' is used in similar ways or stems from similar sources? A *theory of meaning* specifies how meaning is assigned to database elements

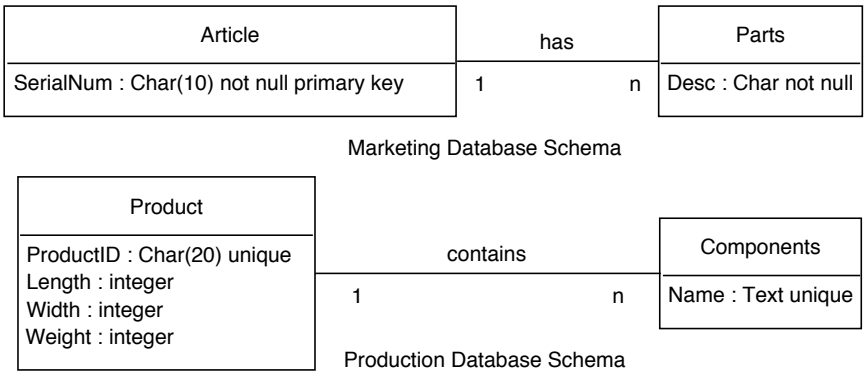


Fig. 1. Example: Matching a marketing to a production management database

and consequently how their similarity of meaning is established.

The performance of a schema matching method is determined by comparing the matches it makes to a set of assumed correct reference matches. Performance is assessed in terms of precision (the proportion of matches found by the method that are in the set of reference matches) and recall (the proportion of reference matches found by the method). These reference matches are in turn based on a human assessment of the databases. Hence, a method is successful when it satisfies two criteria:

(1) The method does the right thing: It embodies the same theory of meaning as that applied by the method users or evaluators in establishing reference correct matches [13].
(2) The method does the thing right: It correctly implements the theory of meaning held by its users or evaluators.

Hence, to improve the performance of schema matching methods, knowledge of human theories of meaning is required in order to correctly attribute method performance either to the first or second aspect and to consequently focus development efforts in the right direction.

To date, existing research has focussed primarily on the second aspect and there is a lack of empirical knowledge of the theories of meaning held by database integrators. Despite the multitude of proposed schema matching methods, no systematic, theory-based, empirical inquiry into the nature of the theories of meaning that database integrators hold has been presented. This paper therefore examines the following research question:

> *What are the theories of meaning that are held by database integrators and to what extent are they applied to schema matching problems?*

An answer to this question can help method developers improve their matching methods by adapting them to human theories of meaning for database matching. This paper presents an experimental study that investigates what theories of meaning are held by database integrators and to what extent they contribute to schema matching.

The remainder of the paper is structured as follows. Section 2 provides a brief introduction to current research in schema matching. Section 3 develops the theoretical basis for this research. Section 4 describes the research design and the development of the experimental stimuli and measurement instruments. Section 5 presents the data collection and sample characteristics, followed by data analysis in Section 6. Sections 7 and 8 discuss implications of the results and limitations of the study, and Section 9 concludes the paper with an outlook to future research.

2 Schema Matching

2.1 Schema Matching Methods

Schema matching methods are primarily categorized by their use of schema-level or instance-level information, although many methods use both types of information [1,2].

Schema-Level Techniques Schema-level matching methods may use constraints information, such as data type constraints, optionality constraints, or uniqueness constraints of attributes [3,14–20]. The use of constraint information assumes that this information is meaningful for judging the similarity of database elements. For example, does the fact that attribute A of table X is of type "character" make it similar to attribute B of table Y, which is of type "text"?

In addition, or as an alternative to, constraint information, linguistic information may be used for matching on the schema level. Linguistic information may be used by measuring character string similarity [14,15,3,5,18] or by using externally supplied dictionaries, thesauri or lexical databases [21,22,4,16,23,11,24,25,19] such as WordNet [26] or CyC [27]. Such linguistic databases provide lists of homonyms, synonyms, and other standard semantic relationships between words.

Linguistics-based techniques are limited to problems where linguistic information is available. For example, a database schema with tables and attributes that possess abbreviated names or acronyms like "TAB-BKHY" and "ATTR-BGHO" offers little linguistic information. Moreover, the semantics that are relevant to the particular matching task are not necessarily those encoded in the externally provided lexicon or thesaurus. For example, a company (and their database schema) may understand *products* as the outcome of a development process that is sold to business customers and supported by service teams, whereas *merchandize* are things that are sourced externally, sold to consumers and supported by the supplier. While WordNet [1] lists these terms as synonyms, they are in fact used in very different ways.

Schema-level matching methods may also use structural information, i.e. the relationships between database elements such as relationship-types between entity-types or foreign-key dependencies between tables [4,5,14,15,23,24,28,19,18]. The use of structural information for identifying matching database elements may be limited to local structures, where relationships only between directly

[1] Version 1.7.1

connected database elements are considered, or it may encompass global structures, where the overall structure of the database is considered [14].

Recent interest in ontologies and XML data has led to matching methods that exploit special properties of hierarchical schema [15,23,11,28,29]. While this helps in reducing the set of possible matches, it also restricts the applicability of these approaches.

Instance-Level Techniques Information about schema instances can be used in addition to or instead of schema information. For example, a schema-level matcher may be used to match entity types and an instance-level matcher may subsequently be used to match attribute [30,7,31–33].

To identify attributes with similar meaning, aggregate instance information such as value distributions, term frequencies, averages, etc. is computed for attributes or columns and then compared across attributes or columns to yield a similarity measure. For example, when two table columns contain the same distribution of values, then the columns are argued to be similar in meaning. Machine learning techniques such as neural networks [33] and Bayesian learners [6,7] among others [30,9] can establish characteristic features of an attribute or column which can then be compared to others.

All the above types of information are used by one matching method or another, often in different combinations. However, without knowledge of the contribution of each type of information to human similarity judgements, the developers of schema matching methods have little guidance on how to improve their methods.

2.2 Method Evaluation

Evaluation of schema matching methods is done by comparing their results to a set of reference matches [3–12]. However, none of the experimental studies explicate the theories of meaning held by subjects or researchers that establish reference matches. The experimental evaluations involving human-based reference matches also do not report the specific task context that subjects were asked to assume when determining the matches, if any. Experimental evaluations frequently suffer from lack of validity. When reference matches are established by a single domain expert, as in [32,4], the method and its performance may reflect the idiosyncrasies of the specific expert and may not be generalizable. When the researchers themselves [34,7,29,35,6] establish the reference matches, it is not surprising that the method achieves a high level of performance, as the method developers have intricate knowledge of the theory

of meaning used to establish the reference matches. However, schema matching researchers may not be representative of data integration professionals. Many studies make no mention at all about how reference maches were established [36,25,16,37,12,23,11]. Only a few studies report the use of multiple subjects to establish reference matches and thereby establish some validity [24,5,10] .

Because of the validity issues surrounding the establishment of reference matches, method performance cannot be attributed with certainty to either of the two aspects pointed out in Section 1. A method's poor performance may be caused either by implementing a different theory of meaning from that of the human subjects establishing the reference matches, or by errors or omissions in the implementation.

3 Theories of Meaning

The problems pointed out in Sections 1 and 2, not being able to assess whether methods implement the right theory of meaning and not being able to offer empirical guidance for their improvement, could be addressed if it was known how humans attach meaning to database elements and how they judge the similarity of meaning of database elements. Theories of meaning play an important role in addressing this question and have a long history in philosophy, psychology, and related disciplines. A multitude of different theories have been discussed in these disciplines. This section can only broadly present the most influential of these theories.

3.1 Feature Theory of Meaning (Constraints)

Frege [38] suggests that the meaning of a term or phrase is its "sense". This "sense" is defined by Russell as a function of the meaning of the logical operators, predicates and referents making up that phrase [39,40]. Consider a database with the following instance: (ProductID=123ABC (and) Length=240mm (and) Width=120mm (and) Weight=5kg). According to this theory of meaning, the meaning of this database instance is a combination of the meaning of the individual predicates (ProductID, Length, Width, Weight), based on the logical operator connecting them (in this case the *and* operator, and their referents (the real world instances denoted by '123ABC', '240mm', '120mm', '5kg').

Schema-based matching methods that match entity-types based on their attributes make implicit use of this theory of meaning. Here, the attributes are features of the entity-types. Similarly, attribute matching based on syntacti-

cal constraints uses this theory. In this case, syntactical constraints, such as datatypes, uniqueness, and others, are features of an attribute.While no proposed method uses this theory of meaning by itself, aspects of it are found in ARTEMIS [5], MUVIS [19] and the proposals in [17,18].

3.2 Denotational Theory (Instances and Aggregates)

The denotational theory of meaning, proposed in different forms [38,40–42], holds that the meaning of a term or phrase is determined by its reference to objects or states of the world. Elements of this theory are found in all instance-based schema matching methods. The meaning of a term, e.g. the name of a database table, is what it denotes. For example, the name of the database table "Product" denotes to all its instances. Consequently, the meaning of two database tables is similar, if their sets of instances are similar [6]. For reasons of efficiency, matching tools generally do not directly compute the number of matching instances, but instead use aggregate information, such as value distributions [7,12,32].

3.3 Early Pragmatist Theory (Effects)

Pragmatist theory [43,44] focuses on the capacity of statements to create effects in the world. Statements are meaningful when there is a correlation between what is said and what consequently happens. Early pragmatist theory [43] is behaviorist in nature; it refers only to the observable effects of the use of statements. In schema-matching, the early pragmatist theory can be exploited by observing the effects of a statement. For example, adding an instance or modifying instance data are akin to making statements. The effect, and therefore the meaning, of statements is the observed behaviour in the database, the application logic, or the organizational real world, after that statement is made. For example, we might observe that in one database, upon creation of a new instance of "Product", new instances of "Components" are created. If we observe, in a second database, that upon creation of a new instance of "Merchandise", new instances of "Component" are created, according to pragmatist theory, "Product" and "Merchandise" have similar meaning.

3.4 Late Pragmatist Theory (Intent)

In contrast to the early behaviorist account of pragmatism, later pragmatist theory [45–47] suggests that the meaning of a statement is not its actual effect but its *intended effect*. To discover the intention of a statement in the database,

e.g. the insertion of an instance of "Product", we must examine what the user intended by making this statement. One way this can be done is by tracing database operations to a particular software application or part of a workflow. It is often easier to identify the user's intent from the software module, rather than the database. For example, assuming some known overlap among the data stored in two databases, when a software module "AddProducts" inserts an instance X into table "Product" in one database and the same instance X into table "Item" in the other database, this may increase our confidence that "Product" and "Item" have the same, or at least similar, meaning. Of course, this operationalization, as well as that for the Early Pragmatist Theory (Section 3.3), requires multiple observations of non-trivial behaviour to increase our confidence. In summary, this theory of meaning suggests that if operations on database elements originate in the same application software modules, the database elements have similar meaning.

3.5 Knowledge Based Theory (Structure)

The knowledge-based theory of meaning [48,49] suggests that words or statements acquire meaning through an underlying theory, which defines their meaning through relationships to other words or statements. For example, the meaning of the term "water" is determined by chemists as the characteristics that whatever is "water" must consist of hydrogen and oxygen, among other criteria. Such a theory is a connected web or network of propositions [50,51]. In the knowledge-based theory of meaning, as well as in the related coherence theory [52–55], two words or phrases have similar meaning, if they stand in similar relationships to other words or phrases. Schema matching methods may exploit the knowledge-based theory of meaning by recognizing that every schema can be seen as a theory or a network of related terms. This requires matching attributes or entities in the context of the entire schema. Elements that occupy similar positions in two schemata, can be assumed to have similar meaning. This theory of meaning has been applied in the ONION method [14].

4 Research Design

This study intends to determine whether, and to what extent, the different theories of meaning described in Section 3 are employed by humans for the task of schema matching. From the review of theories of meaning in Section 3, we have identified the factors in Table 1. While similarity of name does not feature in any theory of meaning, it might play an important role and was included as an additional factor. We are interested in the relative contribution

Factor	Label	Variable Name
Structural Similarity (from knowledge based theory)	STRUC	X_1
Syntactic Constraint Similarity (from feature based theory)	CONST	X_2
Instance Similarity (from denotational theory)	INST	X_3
Similarity of Aggregated Instance Information (from denotational theory)	AGGR	X_4
Similarity of Effects (from pragmatic theory)	EFF	X_5
Similarity of Intent (from pragmatic theory)	INT	X_6
Similarity of Name	NAME	X_7

Table 1
Experimental factors (independent variables)

of these factors to the perceived similarity of meaning of database elements.

We address the research question experimentally by having subjects view information about two fictitious databases and respond with their perception of overall similarity of database elements. We vary the factors in Table 1 in these descriptions and identify how these variations affect the perceived similarity. We only briefly summarize the research design in this section, a more extensive description is found in Appendix A. In this study, each of the seven factors in Table 1 can take on two levels: High similarity and Low similarity. A fractional design [56] is used to reduce the 128 ($= 2^7$) possible experimental conditions to 32, making this study feasible. However, even 32 conditions are too many for one subject, so that we required 4 subjects, each with 8 conditions. To be able to tell subject effects from the manipulation of the seven factors, we replicate this design, thus requiring 8 subjects each working through 8 experimental conditions each. This yields $n = 32$ observations for each factor level.

The remainder of this section presents the detailed methodology of this study. Section 4.1 describes the development of the experimental stimuli, the descriptions of two fictitions databases. To ensure that these are realistic, we interviewed four industry experts. Section 4.2 describes development of questions to measure the perceived similarity of database elements. For this, too we used our four industry experts. We then used a pilot test with undergraduate students (Section 4.3) to make sure that our questionnaire was valid, and that our variations of the factors in Table 1 were good. We then proceeded with the data collection for the actual study, involving eight experienced professionals (described in Section 5).

4.1 Experimental Stimuli

For each experimental condition, stimuli are created by describing aspects of two fictitious databases. The similarity of the different aspects are varied according to the experimental design presented in Appendix A, Table A.1. The variations between high and low similarity should be realistic, to improve the plausibility and believability on the part of subjects [57,58]. At the same time, the varations need to be sufficiently large to enable the subjects to properly discriminate between them. An example of a complete stimulus is shown in Appendix B.

Structural similarity (STRUC) is operationalized as differences in the database schema, i.e. the connections of tables or entity-types to other tables or entity-types. To avoid overly complex stimuli, only a limited number of connections can be shown. The initial stimulus showed only two other tables or entity-types. This was extended to five after discussions with experts, because the difference between similar and non-similar structures was too small.

Constraint similarity (CONST) is operationalized as differences in the data types, uniqueness constraints, and not-null constraints on table columns or attributes. The number of attributes shown to subjects was limited to three in all experimental conditions.

Instance similarity (INST) is operationalized as differences in content of a table or entity-type. Six instances were presented for each database and specifically characterized as examples. Care was taken that the instances conformed to the constraints and datatypes. Realistically, the factors INST and CONST are not completely independent as the data depends in part on the constraints and data types, limiting the possible variations for this study.

Aggregate value similarity (AGGR) is operationalized as differences in aggregate values. For each of the two database elements, we presented the number of rows in each table, the means, and the standard deviations for each table column or attribute. Realistically, the factors AGGR and INST are not completely independent as aggregates depended on the instance values, again limiting the possible variations for this study.

Similarity of effects (EFF) is operationalized by describing the frequency of updates, and inserts, and by describing the effects of deletion and insertion operations on related database elements.

Similarity of intent (INT) is operationalized as differences in the names of the software modules that insert and update information in the two databases.

Following the recommendation in [59], the names of the fictitious database

tables were arbitrary Greek and Hebrew letters, e.g. "Epsilon", "Aleph". This avoids subjects using any implicit meaning they might hold and ensures that only the information that is presented can be used to judge the perceived similarity. Attributes were named by combinations of three letters, again to ensure that subjects cannot rely on their subjective interpretation of words.

To ensure their validity, the initial stimuli were first checked with a faculty member with industry experience in database integration. This led to a number of improvements, such as the inclusion of five instead of two related database elements, the inclusion of uniqueness constraints on attributes, and elaboration of information on the background and purpose of the fictitious exercise in the introductory briefing for subjects.

Next, four senior data integration professionals from two global IT consulting firms, each with more than 5 years of experience, were interviewed about the process of data integration, and the descriptions of the two fictitious databases that were developed. The interviews confirmed that the all the information in the descriptions was relevant and useful for schema matching and database integration. All information provided in the stimuli had been used by these professionals at some point or for some project. As a result of the interviews, changes were made in the introductory briefing for subjects, the graphical depiction of the schema, and minor presentation issues were addressed.

4.2 Measurement Instrument

Developing a questionnaire for the dependent variable, perceived similarity, without over-emphasizing one aspect or another of this construct, proved challenging. The question set included the following questions [variable labels in brackets], measured on a 7-point Likert-type semantic differential scale anchored by "I agree" and "I disagree":

(1) The entities X and Y in the two databases are the same.
(2) The entities X and Y in the two databases have the same meaning.
(3) The entities X and Y in the two databases refer to the same thing.
(4) If I were to integrate the two databases, I would integrate entities X and Y.
(5) The entities X and Y in the two databases store information about the same business objects.
(6) The entity X in the first database matches the entity Y in the second database.
(7) The entity X in the first database corresponds to the entity Y in the second database.

The original question set also included variants of the above items using the terms "certain" and "confident", such as:

- How certain are you that the entities X and Y in the two databases are the same (uncertain/absolutely certain)
- How confident are you that the entities X and Y in the two databases are the same (not at all confident/very confident)

Based on interviews with the four data integration professionals from global IS consulting companies, these variants were dropped from the question set as they implied responsibility and accountability for integration decisions that few professionals would be willing to commit to.

To ensure that all questions are related, a card-sorting exercise was conducted. Four information systems faculty members as subjects were asked to categorize the questions in any way they saw fit. No clear categorization structure was discernible. While all four subjects sorted the questions into two or more categories, there was no agreement on the number of the categories, the items for the different categories, or the labels of the categories that were created. We conclude that the scale consisting of the seven items does in fact measure a single concept, as desired.

4.3 Pilot Test

With preliminary stimulus material from Section 4.1 and the seven questions to measure overall perceived similarity from Section 4.2, a pilot test was conducted with 61 undergraduate students after they had successfully completed a database course. The correlation matrix showed that questions 2 through 7 are all pairwise correlated at $p < .01$. Question 1 is correlated only with item 6 and, not significantly, with question 7.

To determine whether all questions measure a single concept, overall perceived similarity, an exploratory factor analysis using PCA and varimax rotation showed only one component with an eigenvalue > 1 and a scree plot confirmed this. A Maximum-Likelihood (ML) factor analysis also indicated that a single factor solution fits the data well. A confirmatory factor analysis also showed a good model fit with one factor (details in Appendix C, Table C.1). However, the correlation of the first question with the remaining six was low, which led to low factor loadings for the first question, indicating that this question might not be highly related to perceived similarity. Re-analyzing the results with questions two through seven showed a significant improvement.

The pilot test was also used for a manipulation check, to ensure that our variations of the factors between high similarity and low similarity were in fact

perceived as intended. Subjects were asked to rate the perceived similarity of each of the seven factors in the different conditions on a 7-Point Likert-scale. A MANOVA with these ratings as dependent variables and our intended manipulation of the factors as independent variables yielded significant, but confounded, influences:

- Manipulation of STRUC had an effect also on perceived similarity of name (NAME).
- Manipulation of CONST had an effect on perceived similarity of constraints (CONST), instances (INST), and aggregates (AGGR).
- Manipulation of INST did not have an effect on perceived similarity of instances (INST).
- Manipulation of AGGR also had an effect on perceived similarity of name (NAME).
- Manipulation of EFF had an effect on perceived similarity of effects (EFF) and intent (INT).
- Manipulation of INT had no effect.
- Manipulation of NAME had an affect on perceived similarity of name (NAME) and effect (EFF).

A discriminant analysis was conducted to examine whether the perceived manipulations were able to discriminate between the intended manipulations. The results confirmed the MANOVA analysis. The resulting discriminant functions can distinguish among the intended manipulations except for the factors CONST and EFF.

Based on the pilot-test results, the stimuli were adjusted to strengthen their visibility and to increase the differences between the similar and not-similar conditions. Changes included increasing or decreasing the values given for the AGGR operationalization to increase the differences between similar/non-similar conditions, changing the values of the INST operationalization to more clearly distinguish between the similar/non-similar conditions, separating attribute descriptions from the schema diagram to a separate page in order to separate CONST from STRUC manipulation, including additional fictitious names of software modules for the INT and EFF manipulations, and separating the tables for INST and AGGR.

Subjects were also asked to report problems with the questionnaires and recommendations for improvement. Many subjects (13 of 15 responses) responded that the questions were very similar (which was desired, as they are all intended to measure perceived similarity) and recommended to use more informative names (which was not done for reasons described above). No other problems were reported.

Subject	Business Context	Experience		Job Title	Education
		Years	Projects		
a	Application Integration	3	3	Database Administrator	Information Systems
b	Data rationalization and sharing	5	3	Database Administrator	Business and Computer Science
c	Date warehousing	3	5	IT Manager	Computer Science
d	Data Modelling, Data Design	20	10	Senior Principal Software Engineer	
e	Application Integration		2	Technical Designer	PhD Computer Science
f	Data warehousing/business intelligence	8	9	Principal Consultant	BSc Computer Science
g	Information Integration			Program director information integration	
h				Database Administrator	Computer Science

Table 2
Self-reported Subject characteristics

5 Data Collection

Section 4 described how our experimental materials and questions were tested for validity. Printed questionnaires were assembled and sent to eight data integration professionals for two replications of the experimental design (Table 2). Subjects were randomly assigned to a block of the experimental design presented in Section 4 and Appendix A, Table A.1. The full questionnaire consisted of an introductory briefing, two experimental conditions as a manipulation check, eight experimental conditions according to the assigned block, demographic questions and the following three qualitative questions:

- What are the three most important reasons for matching two database elements?
- What are the three least important reasons for matching two database elements?
- Name any information that is useful for matching database elements that is not on this questionnaire.

Each of the two manipulation check conditions was followed by a set of questions asking subjects to rate the perceived similarity of the 7 manipulated factors, analogous to the questions on the pilot test. Each of the eight condi-

Subject	t-test significance
a	0.0300
b	0.0032
c	0.0000
d	0.0000
e	0.0000
f	0.0005
g	0.0761
h	0.0000

Table 3
Manipulation check

tions of the assigned experimental block was followed by the set of questions measuring perceived similarity, as described in Section 4.

6 Data Analysis

The first step in the data analysis is to again ensure the validity and reliability of the question set for the dependent variable, perceived similarity. Unidimensionality of the scale was again confirmed using PCA with varimax. The eigenvalue criterion and scree plot indicated a single component. Factor analysis with ML extraction of one factor showed good data fit (details in Appendix C, Table C.3). A SEM analysis using a single factor model, also showed good fit. However, given the low correlations of the first question (Appendix C, Table C.2) and the low factor loadings of this question, we re-analyzed a six-item scale and found significant improvement (details in Appendix C). The ML extraction of one factor from a six-item set was then used to derive regression scores of perceived similarity for further analysis (variable "PSim").

A one-tailed t-test was conducted on the similarity ratings of the manipulation check experimental conditions. The t-tests showed significant differences at the $\alpha < 0.1$ level between the low and high manipulation conditions for all subjects (Table 3), indicating that manipulation of the conditions was successful: All subjects recognized intended high similarity as high and intended low similarity as low.

The data was analysed using a repeated-measures (also called "within-subject" or "split-plot") ANOVA procedure with INT and NAME and their interaction as the between-subject variables ("whole plot") and with STRUC, CONST, INST, AGGR, and EFF and their two-factor interactions as the within-subject variables (Refer to Section 4 and Appendix A for details). The ANOVA results are shown in Table 4. While we have shown the unidimensionality of the

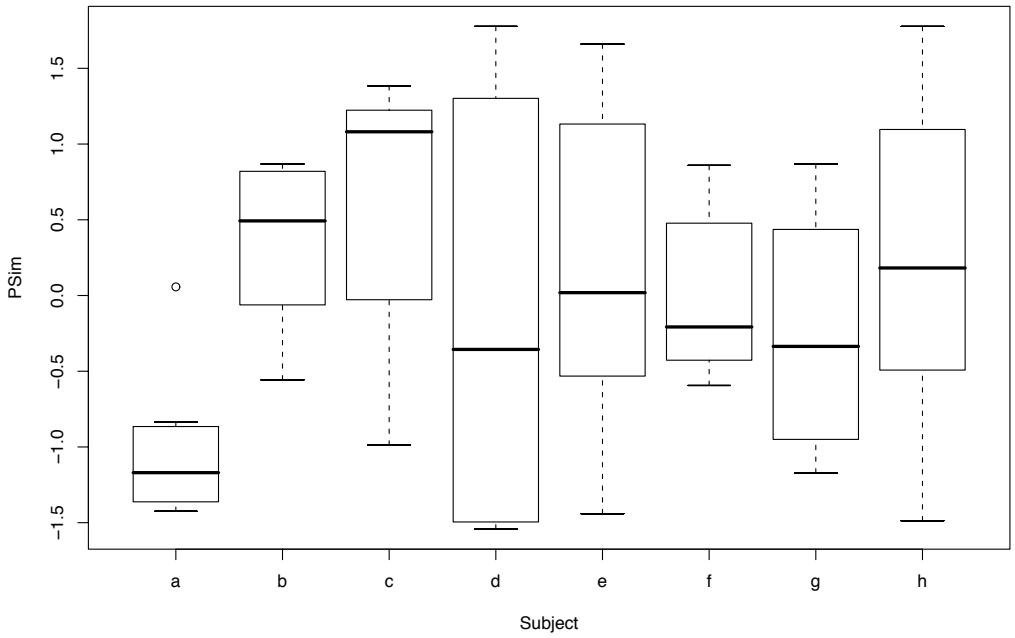


Fig. 2. Factor plot for Subject

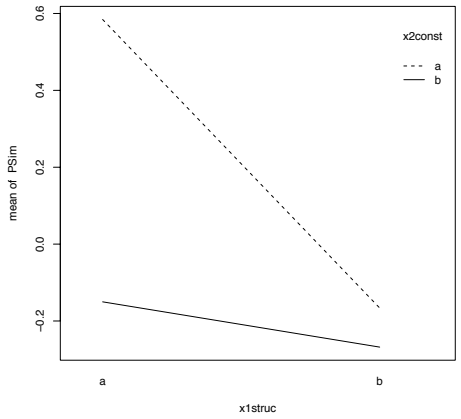
dependent variable both in the pilot test and in the actual test data, we confirmed the ANOVA results using a MANOVA analysis with the seven questions of the similarity measure as dependent variables. The results, using Wilks' Λ test, confirmed the ANOVA results for significant effects. The demographic variables we collected, years of experience and number of projects (Table 2), have too many missing values to include them in the analysis as explanatory variables.

Five of the seven factors had a significant main effect on perceived similarity. Three of the two-factor interactions (INT:AGGR, INT:EFF, STRUC:CONST) had a significant effect, as did two subject – factor interaction effects (STRUC:Subject, INST:Subject), suggesting that not only is there relatively large variability of perceived similarity within subjects (as evidenced by the error stratum for subjects), but subjects also differ in their reaction to manipulation of structure and instance data. The factor plot for the subject term in Fig. 2 clearly shows this between-subject variability and the value of a within-subjects analysis.

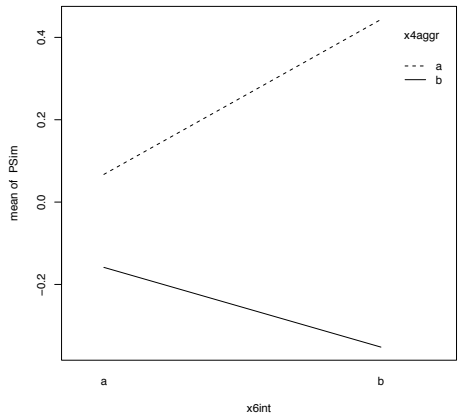
The interaction plots in Fig. 3 show the following phenomena. In Subfigure (a), the lower, solid line shows that when the constraints are not similar (CONST="b"), there is little effect of structure (STRUC) on perceived similarity. Only when constraint information is similar (CONST="a", upper, dashed line), does perceived similarity decrease as the structural similarity is decreased (STRUC="b").

Error: Subject						
	Df	Sum Sq	Mean Sq	F value	Pr(>F)	Sig
x6int	1	0.1333	0.1333	0.0469	0.8392	
x7name	1	0.4290	0.4290	0.1508	0.7176	
x6int:x7name	1	1.7328	1.7328	0.6091	0.4787	
Residuals	4	11.3801	2.8450			
Error: Within						
	Df	Sum Sq	Mean Sq	F value	Pr(>F)	Sig
x1struc	1	3.0163	3.0163	11.0000	0.0046960	**
x2const	1	2.7974	2.7974	10.2017	0.0060374	**
x3inst	1	9.4001	9.4001	34.2811	3.166e-05	***
x4aggr	1	4.1735	4.1735	15.2204	0.0014174	**
x5eff	1	5.6196	5.6196	20.4942	0.0004011	***
x6int:x1struc	1	0.0338	0.0338	0.1231	0.7305828	
x6int:x2const	1	0.1471	0.1471	0.5364	0.4751979	
x6int:x3inst	1	0.0642	0.0642	0.2342	0.6354029	
x6int:x4aggr	1	1.2989	1.2989	4.7369	0.0459102	*
x6int:x5eff	1	2.1616	2.1616	7.8833	0.0132517	*
x7name:x1struc	1	0.0165	0.0165	0.0603	0.8093789	
x7name:x2const	1	0.1501	0.1501	0.5473	0.4708574	
x7name:x4aggr	1	0.0143	0.0143	0.0521	0.8224645	
x1struc:x2const	1	1.6017	1.6017	5.8411	0.0288587	*
x1struc:x3inst	1	0.1535	0.1535	0.5599	0.4658770	
x1struc:x4aggr	1	0.0001	0.0001	0.0003	0.9866967	
x1struc:x5eff	1	0.0560	0.0560	0.2044	0.6576921	
x1struc:Subject	5	4.6365	0.9273	3.3818	0.0303658	*
x2const:x3inst	1	0.0297	0.0297	0.1082	0.7467886	
x2const:x4aggr	1	0.6684	0.6684	2.4375	0.1393143	
x2const:x5eff	1	0.0741	0.0741	0.2702	0.6108075	
x2const:Subject	5	0.7584	0.1517	0.5532	0.7338886	
x3inst:x4aggr	1	0.0051	0.0051	0.0187	0.8929897	
x3inst:Subject	4	3.9422	0.9855	3.5942	0.0302003	*
x4aggr:x5eff	1	0.0001	0.0001	0.0002	0.9887285	
x4aggr:Subject	5	1.3604	0.2721	0.9922	0.4549611	
Residuals	15	4.1131	0.2742			
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' 1'						

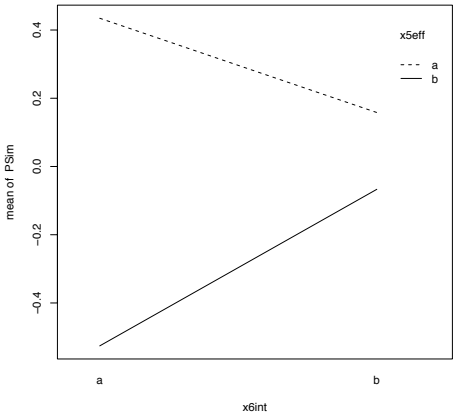
Table 4
ANOVA Results



(a) Structure and Constraints



(b) Intention and Aggregate Data

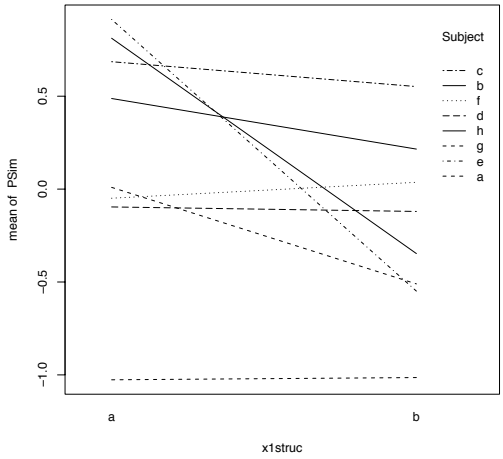


(c) Intention and Effects

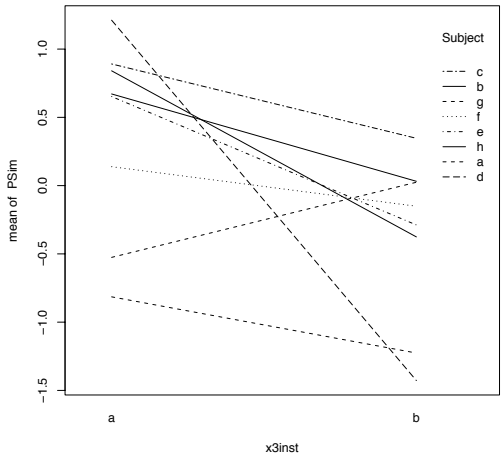
Fig. 3. Interaction Plots ('a'=High similarity, 'b'=Low similarity)

Subfigure 3 (b) shows a counterintuitive effect that when the aggregate information is similar (AGGR="a", upper, dashed line), perceived similarity rises as similarity of intent drops (INT="b"). However, when the aggregate information is dissimilar (AGGR="b", lower, solid line) we see the expected effect that as similarity of intent decreases (INT="b"), so does overall perceived similarity.

Subfigure 3 (c) shows that when the similarity of effects is low (EFF="b", lower, solid line), similarity of intent (INT) has a counterintuitive negative effect on perceived similarity: perceived similarity rises as the similarity of intent decreases. However, when similarity of effects is high (EFF="a", upper, dashed line), similarity of intent has a the expected positive effect: perceived similarity is high when similarity of intent is high (INT="a") and low when similarity of intent is low (INT="b").



(a) Subjects and Structure



(b) Subjects and Instances

Fig. 4. Subject Interaction Plots ('a' = High similarity, 'b' = Low similarity)

Figure 4 shows the significant interaction effects of subjects and the controlled factors. Subfigure (a) shows that only two subjects ("e" and "h") show the expected effect and rate perceived similarity low when structural similarity is low (STRUC="b"), while the remaining subjects show little or no effect to manipulation of structural similarity. Subfigure (b) shows that subjects tend to rate perceived similarity lower if instance information (INST) is manipulated to be lower (INST="b"), an expected effect. However, one subject ("g") shows an inverse effect.

As our research question asked about the relative effect size of each factor on the perceived similarity of the database elements, we examine two measures of effect size for the significant factors. The most widely used measure is Cohen's

Effect	Effect Size Estimate	
	Cohen's f	ω^2
INST	0.72	0.34
EFF	0.55	0.23
AGGR	0.47	0.18
STRUC	0.39	0.14
CONST	0.38	0.13
INT:EFF	0.33	0.10
STRUC:CONST	0.28	0.07
INT:AGGR	0.24	0.06

Table 5
Effect sizes for significant effects

f, whose estimate is proportional to the F-Ratio of an effect:

$$f = \sqrt{\left[\frac{df}{N}\right](F-1)}$$

where df are the degrees of freedom, F is the ANOVA F-test statistic and N is the number of observations. Cohen's f is directly related to another widely used measure of effect size, ω^2:

$$\omega^2 = f^2/(1+f^2)$$

Table 5 shows the effect sizes for the significant effects in the present study. An effect size of $f > 0.4$ is considered a large effect [60]. We can see that the main effects are large; INST has the largest effect, followed by EFF, AGGR, STRUC, and CONST. The interaction effects INT:AGGR and INT:EFF are medium size effects and the CONST:AGGR interaction effect is small. The main effect sizes are also evident in the design plot in Figure 5.

The answers to the three qualitative questions were tabulated and ranked by number of mentionings, shown in Tables 6, 7, and 8. Noticeable is the perceived importance of structural information over content information. This perception is in contrast to the actual behaviour, where INST had a greater effect than STRUC. The ranking of aggregate information near the bottom of both lists is also in contrast to actual behaviour, where AGGR had the second largest effect.

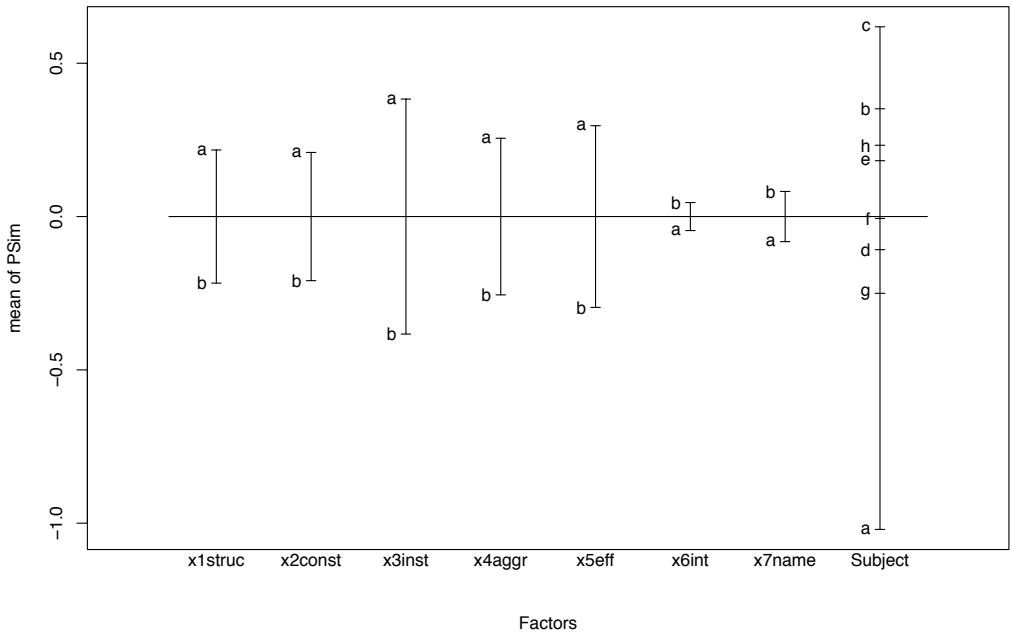


Fig. 5. Main Effects ('a' = High similarity, 'b' = Low similarity)

5	Similarity of relationships	STRUC
3	Similarity of changes to the data	EFF
2	Similarity of content	INST
2	Similarity of data usage by application software	INT
1	Similarity of primary keys	CONST
1	Similarity of data types	CONST
1	Similarity of aggregates	AGGR

Table 6
Most important reasons for matching database elements

7	Dissimilarity of relationships	STRUC
4	Dissimilarity of content	INST
2	Dissimilarity of data usage by application software	INT
2	Dissimilarity of aggregates	AGGR
2	Dissimilarity of application software effects	INT
2	Dissimilarity of attributes and data types	CONST
1	Dissimilarity of name	NAME

Table 7
Most important reasons for NOT matching database elements

2	Source of data
1	Business context
1	Business use of data
1	Business domain model
1	Business rules for data
1	Semantics of data
1	Actual names of entities and attributes

Table 8
Useful information missing from the questionnaire

7 Discussion and Implications

The experimental results show a number of important findings with implications for future research and applications. First, there is no single contributor to perceived similarity. Table 5 shows the size of effect, and consequently the importance for schema matching of the factors we investigated. We see that similarity of instance information had the largest effect by far on on overall perceived similarity ($f = .72$), followed by similarity of effects ($f = .55$), and similarity of aggregates ($f = .47$). Less important are structural ($f = .39$) and constraint ($f = .38$) similarity. Three interaction effects are also statistically relevant, however the size of their effect is minor in comparison ($.24 \leq f \leq .33$).

The implication for schema matching applications is the importance of using multi-method approaches to identifying similar database elements. Work on this has begun with the GLUE method [29,35,6] and the SEMINT method [61,33], both of which employ multiple matchers in combination. In this context, the effect sizes in Table 5 may be used as weightings for aggregating similarity estimates from different schema matching methods. For example, the overall similarity of a two database schemata may be computed as 0.72 times the measured similarity of instances plus 0.55 times the measured similarity of database effects, etc. While we believe this is already an important result, more detailed analyses are necesary to determine how humans assess for example the similarity of instances or the similarity of structure.

Second, the effect sizes for instance-based effects are larger than those for structure-based effects, indicating that instance information is more important to schema matching than structural information. The two largest effects are those of instance (INST) and aggregate information (AGGR), while structural information (STRUC) and constraints and data types (CONST) have effects that are only half as large as the INST effect, indicating these are only half as important. Given this importance of instance based approaches, research emphasis should focus on exploiting instance information to the best possible extent. A promising start in this direction are feature learning approaches

[34,7], which aim to select the most relevant features of database instances for the matching.

In connection with this, we found that while the observed importance of instance information was much greater than that for structural information, the self-reported measures in Tables 6 and 7 showed the inverse. One possible explanation is the briefing sheet, which indicated that the databases covered the same domain. This may have led subjects to assume that the databases contain the same instances and led to the importance of instance information in matching. However, a more likely explanation is a systematic bias of the self-report measures for which a variety of highly plausible reasons exist [62].

Third, the subject–factor interaction effects show that, while its importance is relatively low as indicated by its effect size (Table 5), care must be taken when generalizing the applicability and appropriateness of schema matching methods. For example, not all subjects perceive structural similarity to be a positive contributor to overall perceived similarity (Fig. 4 (a)). Schema matching methods must be applied with awareness of the notions of similarity held by their users. This is especially important in the validation of the heuristics. What is satisfactory or useful to the method developer, may not be satisfactory or useful to a user. This requires that the establishment of reference matches for empirical evaluations must take into account the nature of the theory of meaning held by the user.

Fourth, the variations between subjects imply that the effect sizes are averages and may vary between different users of schema matching methods. Instead of a "one size fits all" method, methods may need to be tailored to suit specific users. Here, learning based matching methods [33,6,7,30] may be at a disadvantage if they learn from different users. Instead, the results learned from different users should be kept separate, as the users may hold different theories of meaning.

Fifth, the significant effect of information about database effects (EFF) on perceived similarity points to the need for more research in this area. Currently, none of the proposed schema matching methods make use of information about database effects.

Sixth, from the comparison of the perceived importance of various factors and the actual behaviour, it is evident that these do not always coincide. For example, structural similarity was rated very highly, but its actual effect was much lower than that of instance and aggregate information. Consequently, future research needs to focus on experimentally determining the actual importance of factors to subjects, instead of soliciting their perception or opinion.

Finally, the list of missing information in Table 8 suggests future extensions to schema matching methods. For example, business rules for data are often

embodied in triggers and procedures in databases. This information could be included in future schema matching methods. Thieme and Siebes [63,64] make some use of such behavioural information in the context of matching object-oriented schemata.

8 Future Research

While this study has shown important results and implications for the improvement of schema matching methods (Section 7), as an exploratory study it was designed for breadth of coverage, rather than a detailed understanding of one particular aspect of schema matching. Consequently, there are many opportunities for future research. For example, we have focused only on matching of entities or tables, while realistic schema matching also include matching of attributes or columns. Furthermore, we have assumed a simple one–to–one mapping of entities of the two databases, while schema matching may also include many–to–one and one–to–many matches. Realistic databases also include active features such as triggers, stored procedures, and user–defined functions [65] which have not been considered here. There may be context or situational factors, such as the purpose of the database integration or the implementation technology for the integration, that affect the relative importance of the investigated factors. There may also be subject specific factors, such as experience with different database software and tools, that could affect how humans perceive the similarity of database elements. However, as the number of experimental conditions, and therefore the required number of expert participants, increases exponentially with the number of factors under investation, these aspects are left for separate future studies.

This study neglects the role of domain knowledge. This is intended as the aim of the study is to improve schema matching software, which has no recourse to domain knowledge. However, as matching methods are supplemented with domain ontologies, future studies may find it worthwhile to investigate how this is best done to ensure conformity to the way in which humans make use of their domain knowledge. Interestingly, only one of our respondents suggested that names of tables and attributes were missing but would be helpful (Table 8). This may indicate that element names, and the domain knowledge linked to them, is not as important as it might appear. Perhaps database integrators realize that names may be rather arbitrary and not necessarily an indication of shared meaning. Certainly in this study similar names were not construed to indicate similar meaning, as there was no significant effect of the naming factor.

Another question, already raised in Section 7, is that of measuring the similarity of each aspect of our databases in more detail. For example, what are

the specific factors that determine how people perceive structural similarity, or factors that determine similarity of instances. The effects found to be significant in this study (Table 5) should be examined in more detail. For example, structural information may be separated into relationships between database elements, cardinalities of the relationships, and constraints on the foreign keys. These can then be manipulated separately.

9 Conclusion

This paper presents the first systematic empirical study of what information is useful for schema matching. An experiment was conducted to determine the theories of meaning held by data integration professionals. Based on five theories of meaning, 32 experimental stimuli were constructed and tested on eight subjects. We determined the relative importance of seven kinds of information and found that the most important effect on the perceived similarity of schema elements was that of instance and aggregate information, while structure and constraint based information had only half as large an impact on perceived similarity. From these findings, we have derived seven specific recommendations for the improvement of schema matching methods and pointed out areas for future research.

References

[1] C. Batini, M. Lenzerini, S. Navathe, A comparative analysis of methodologies for database schema integration, ACM Computer Surveys 18 (4).

[2] E. Rahm, P. A. Bernstein, A survey of approaches to automatic schema matching, The VLDB Journal - The International Journal on Very Large Databases 10 (4) (2001) 334–350.

[3] B. S. Lerner, A model for compound type changes encountered in schema evolution, ACM Transactions on Database Systems 25 (1) (2000) 83–127.

[4] P. Z. Yeh, B. Porter, K. Barker, Using transformations to improve semantic matching, in: Proccedings of K-CAP'03, Sanibel Island, FL, 2003, pp. 180–189.

[5] L. Palopoli, D. Sacca, G. Terracina, D. Ursino, Uniform techniques for deriving similaritites of objects and subschemas in heterogeneous databases, IEEE Transactions on Knowledge and Data Engineering 15 (2) (2003) 271–294.

[6] A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, A. Halevy, Learning to match ontologies on the semantic web, The VLDB Journal 12 (2003) 303–319.

[7] J. Berlin, A. Motro, Database schema matching using machine learning with feature selection, in: Proceedings of CAISE 2002, Toronto, ON, 2002, pp. 452–466.

[8] J. Kang, J. F. Naughton, On schema matching with opaque column names and data values, in: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, 2003, pp. 205–216.

[9] X. Su, S. Hakkarainen, T. Brasethvik, Semantic enrichment for improving system interoperability, in: Proceedings of the 19th ACM Symposium on Applied Computing (SAC 04), Nicosia, Cyprus, 2004.

[10] X. Su, J. A. Gulla, An information retrieval approach to ontology mapping, Data and Knowledge Engineering 58 (1) (2006) 47–69.

[11] N. Noy, M. Musen, The PROMPT suite: Interactive tools for ontology merging and mapping, International Journal of Human-Computer Studies 59 (6) (2003) 983–1024.

[12] W. W. Cohen, Data integration using similarity joins and a word-based information representation language, ACM Transactions on Information Systems 18 (3).

[13] M. Benerecetti, P. Bouquet, S. Zanobini, Soundness of schema matching methods, in: Proceedings of the European Semantic Web Conference 2005, 2005, pp. 211–225.

[14] P. Mitra, G. Wiederhold, M. Kersten, A graph-oriented model for articulation of ontology interdependencies, in: Proceedings of the 7th International Conference on Extending Database Technology EDBT, 2000, pp. 86–100.

[15] E. Bertino, G. Guerrini, M. Mesiti, A matching algorithm for measuring the structural similarity between an XML document and a DTD and its applications, Information Systems 29 (2004) 23–46.

[16] S. Castano, V. De Antonellis, S. De Capitani di Vimercati, Global viewing of heterogeneous data sources, IEEE Transactions on Knowledge and Data Engineering 13 (2) (2001) 277–297.

[17] J. Larson, S. Navathe, R. Elmasri, A theory of attribute equivalence in databases with application to schema integration, IEEE Transactions on Software Engineering 15 (4) (1989) 449–463.

[18] W. Gotthard, P. C. Lockemann, A. Neufeld, System-guided view integration for object-oriented databases, IEEE Transactions on Knowledge and Data Engineering 4 (1).

[19] S. Hayne, S. Ram, Multi-user view integration system (MUVIS): an expert system for view integration, in: Proceedings of the sixth international conference on data engineering, 1990, pp. 402–409.

[20] S. Spaccapietra, C. Parent, View integration: A step forward in solving structural conflicts, IEEE Transactions of Knowledge and Data Engineering 6 (2).

[21] S. Bergamaschi, S. Castano, M. Vincini, Semantic integration of semistructured and structured data sources, ACM SIGMOD Record 28 (1) (1999) 54–59.

[22] R. Lawrence, K. Barker, Integrating relational database schemas using a standardized dictionary, in: Proceedings of the 2001 ACM Symposium on Applied computing, 2001, pp. 225–230.

[23] N. Noy, M. Musen, Anchor-PROMPT: Using non-local context for smeantic matching, in: Workshop on Ontologies and Information Sharing at the 7th International Joint Conference on Artificial Intelligence IJCAI, Seattle, WA, 2001.

[24] S. Melnik, H. Garcia-Molina, E. Rahm, Similarity flooding: A versatile graph matching algorithm and its application to schema matching, in: Proceedings of the 18th International Conference on Data Engineering (ICDE'02), 2002.

[25] M. Bright, A. Hurson, S. Pakzad, Automated resolution of semantic heterogeneity in multidatabases, ACM Transactions on Database Systems 19 (2) (1994) 212–253.

[26] C. Fellbaum, WordNet: An Electronic Lexical Database, The MIT Press, Cambridge, MA, 1998.

[27] D. B. Lenat, CYC: A large-scale investment in knowledge infrastructure, Communications of the ACM 38 (11) (1995) 33–38.

[28] T.-L. J. Wang, K. Zhang, K. Jeong, D. Shasha, A system for approximate tree matching, IEEE Transactions on Knowledge and Data Engineering 6 (4) (1994) 559–571.

[29] A. Doan, P. Domingos, A. Y. Halevy, Reconciling schemas of disparate data sources: A machine-learning approach, in: Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, 2001, pp. 509–520.

[30] R. J. Miller, M. A. Hernandez, L. M. Haas, L. Yan, C. H. Ho, R. Fagin, L. Popa, The Clio project: Managing heterogeneity, ACM SIGMOD Record 30 (1) (2001) 78–83.

[31] J. Kang, J. F. Naughton, On schema matching with opaque column names and data values, in: Proceedings of SIGMOD 2003, June 9–12, 2003, San Diego, CA, 2003, pp. 205–216.

[32] C. E. H. Chua, R. H. Chiang, E.-P. Lim, Instance-based attribute identification in database integration, The VLDB Journal 12 (3) (2003) 228–243.

[33] W.-S. Li, C. Clifton, S.-Y. Liu, Database integration using neural networks: Implementation and experiences, Knowledge and Information Systems 2 (1) (2000) 73–96.

[34] J. Berlin, A. Motro, Autoplex: Automated discovery of content for virtual databases, in: Proceedings of CoopIS'01, 2001, pp. 108–122.

[35] A. Doan, J. Madhavan, P. Domingos, A. Halevy, Learning to map between ontologies on the semantic web, in: Proceedings of WWW2002, Honolulu, HI, 2002, pp. 662–673.

[36] S. Bergamaschi, S. Castano, M. Vincini, D. Beneventano, Semantic integration of heterogeneous information sources, Data and Knowledge Engineering 36 (2001) 215–249.

[37] W. W. Cohen, Integration of heterogeneous databases without common domains using queries based on textual similarity, in: Proceedings of the 1998 ACM SIGMOD International conference on Management of Data, 1998, pp. 201–212.

[38] G. Frege, On sense and reference, in: P. Geach, M. Black (Eds.), Translations from the Philosophical Writings of Gottlob Frege, 3rd Edition, Blackwell Publishers, 1980, pp. 56–78.

[39] B. Russell, On denoting, Mind 14 (1905) 479–493.

[40] B. Russell, Descriptions and incomplete symbols, in: Logic and Knowledge: Essays, 1901-1950, Allen and Unwin, London, 1956.

[41] A. Tarski, The concept of truth in formalized languages, in: Logic, Semantics, Metamathematics, 2nd Edition, Hacket, Indianapolis, IN, 1935, pp. 152–278.

[42] P. Strawson, On referring, Mind 59 (1950) 320–344.

[43] L. Wittgenstein, Philosophical Investigations, Prentice Hall, Englewood Cliffs, 1953.

[44] J. Dewey, Experience and Nature, Dover, New York, 1958.

[45] J. Austin, How to do things with words, Harvard University Press, Cambridge, MA, 1962.

[46] H. Grice, Meaning, in: Studies in the Way of Words, Harvard University Press, Cambridge, MA, 1989.

[47] J. Searle, Speech Acts, Cambridge University Press, Cambridge, UK, 1969.

[48] S. Kripke, Naming and Necessity, Blackwell, Oxford, 1980.

[49] H. Putnam, The meaning of meaning, in: Mind, Language and Reality: Philosophical Papers, Vol. 1, Cambridge University Press, Cambridge, UK, 1975.

[50] W. v. O. Quine, Two dogmas of empiricism, in: From a Logical Point of View, Harvard University Press, Cambridge, MA., 1953.

[51] I. Lakatos, Philosophical papers, Cambridge University Press, Cambridge, NY, 1978.

[52] D. Davidson, A coherence theory of truth and knowledge, in: E. LePore (Ed.), Truth and Interpretation, Perspectives on the Philosophy of Donald Davidson, Basil Blackwell, Oxford, UK, 1986.

[53] H. Putnam, Reason, Truth and History, Cambridge University Press, Cambridge, UK, 1981.

[54] N. Rescher, The Coherence Theory of Truth, Oxford University Press, Oxford, UK, 1973.

[55] J. Young, A defence of the coherence theory of truth, The Journal of Philosphical Research 26 (2001) 89–101.

[56] G. E. Box, W. G. Hunter, J. S. Hunter, Statistics for Experimenters, John Wiley and Sons, New York, NY, 1978.

[57] P. E. Green, V. Srinivasan, Conjoint analysis in consumer research: Issues and outlook, The Journal of Consumer Research 5 (2) (1978) 103–123.

[58] P. E. Green, V. Srinivasan, Conjoint analysis in marketing: New developments with implications for research and practice, Journal of Marketing 54 (4) (1990) 3–19.

[59] J. Parsons, L. Cole, What do the pictures mean? Guidelines for experimental evaluations of representation fidelity in diagrammatical conceptual modeling techniques, Data and Knowledge Engineering 55 (2005) 327–342.

[60] J. Cohen, Statistical Power Analysis for the Behavioral Sciences, 2nd Edition, Academic Press, New York, NY, 1988.

[61] W.-S. Li, C. Clifton, SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks., Data and Knowledge Engineering 33 (2000) 49–84.

[62] E. M. Hufnagel, C. Conca, User response data: The potential for errors and biases, Information Systems Research 5 (1) (1994) 48–73.

[63] C. Thieme, A. Siebes, Schema integration in object-oriented databases, in: Proceedings of the International Conference on Advanced Information Systems Engineering CAiSE, 1993, pp. 54–70.

[64] C. Thieme, A. Siebes, Guiding schema integration by behavioural information, Information Systems 20 (4) (1995) 305–316.

[65] N. W. Paton, O. Diaz, Active database systems, ACM Computing Surveys 31 (1) (1999) 63–103.

[66] J. D. Carroll, P. E. Green, Psychometic methods in marketing research: Part 1, conjoint analysis, Journal of Marketing Research 32 (4) (1995) 385–391.

[67] R. E. Kirk, Experimental Design: Procedures for the Behavioral Sciences, Brooks/Cole Publishing Company, Belmont, California, 1968.

[68] A. Garcia-Diaz, D. T. Phillips, Principles of Experimental Design and Analysis, Chapman and Hall, London, UK, 1995.

[69] J. L. Myers, A. D. Well, Research Design and Statistical Analysis, 2nd Edition, Lawrence Erlbaum Associates, Mahwah, NJ, 2003.

[70] K. Backhaus, B. Erichson, W. Plinke, R. Weiber, Multivariate Analysemethoden, 8th Edition, Springer Verlag, Berlin, Germany, 1996.

[71] J. E. Hair, R. E. Anderson, R. L. Tatham, W. C. Black, Multivariate Data Analysis, 5th Edition, Prentice Hall, Upper Saddle River, 1998.

[72] J. H. Steckel, W. S. DeSarbo, V. Mahajan, On the creation of acceptable conjoint analysis experiemtnal designs, Decision Sciences 22 (2) (1991) 435–442.

[73] W. F. Kuhfeld, R. D. Tobias, M. Garatt, Efficient experimental design with marketing research applications, Journal of Marketing Research 31 (4) (1994) 545–557.

A Research Design

This appendix presents details of the experimental design. Two suitable methods for this study are the conjoint analysis [66,57,58], and the analysis of variance (ANOVA) [67,56,68,69]. Both methods can provide an estimate of the size of the effect of a factor on the dependent variable. In conjoint terms this is called the part-worth, while in ANOVA terms this is the effect size (typically expressed as Cohen's f [60] or ω^2). Conjoint analysis is essentially a fractional factorial design using a single fraction run entirely within a single subject [70,71]. As conjoint analysis is a resolution II design, it can determine only the main effects of factors. In contrast, ANOVA techniques can also determine factor interaction effects. Because this research is exploratory in nature, we cannot rule out interactions of two factors. Consequently, we choose the ANOVA method.

A.1 Fractional Design

Especially in a within-subject experimental design where subjects are assigned to multiple experimental conditions, it is important that all factors have the same number of levels, as subjects would otherwise be biased in their importance estimate: A larger number of distinctions may lead them to believe the factor is more important [58]. Further, because effect size estimation is unproblematic only for two-level factors [68], each of the factors in this study is assigned two levels. Because all factor combinations are plausible and admissible, a regular fractional design can be used [57,72,73].

In any multi-factor factorial design, there is a trade-off between resolution (the ability to uniquely attribute effects to their sources) and experimental

effort. In this case, a factorial design with 7 two-level factors requires $2^7 = 128$ experimental conditions. We neglect interaction effects of three or more factors because of the exploratory nature of the study (we opt for breadth rather than depth), because of their unlikely significance, and because they are difficult to interpret in theoretical terms. Limiting the resolution of the design to the identification of main factor effects and two-factor interaction effects requires estimation of only $7 + 20$ parameters. Hence, we are able to employ a quarter-fraction design with $2^5 = 2^{7-2} = 32$ conditions.

In a fractional factorial design, some of the orthogonal contrasts of higher-order factor interactions in the incomplete design are assigned to main factors of the original design ("generators"). In this case, $7 - 5 = 2$ factors of the complete design must be expressed by the contrasts of higher-order interactions. Depending on which higher-order interaction contrasts of the fractional design are used to express the main factors of the original design, different effects will be confounded, i.e. it becomes impossible to uniquely attribute effects to factors or factor interactions. Choosing a good fractional design requires the identification of generators that will confound main factor effects only with interaction effects of higher order, which are assumed small or negligible. In general, in designs of resolution r, no q-factor interaction effect is confounded with any other interaction effect of fewer than $r - q$ factors. Thus, a high-resolution design is desirable. With the factors labelled as in Table 1, we choose two generators for the resolution IV quarter-fraction design from [56, Table 12.15]: $X_6 = X_1X_2X_3X_4$, $X_7 = X_1X_2X_4X_5$. Table A.1 shows the contrasts assigned to the factors, where "+" indicates similarity of the factor in that condition and "-" indicates dissimilarity.

The *alias structure* of a design shows which factor effects or factor interaction effects are confounded. The alias structure is determined by the generators through the *identity equations* of the design. For the two generators, $X_6 = X_1X_2X_3X_4$ and $X_7 = X_1X_2X_4X_5$ of the 2^{7-2} design, the identity equations are the following:

$$\begin{aligned} X_6 = X_1X_2X_3X_4 &\Leftrightarrow I_1 = X_1X_2X_3X_4X_6 \\ X_7 = X_1X_2X_4X_5 &\Leftrightarrow I_2 = X_1X_2X_4X_5X_7 \\ I_3 = I_1I_2 &\Leftrightarrow I_3 = X_3X_5X_6X_7 \end{aligned}$$

From this, the alias structure can be determined by "multiplying" factors and factor interactions with the identity equations [68,56]. For example, for factor X_1, we find the following confounds:

Num	X_1	X_2	X_3	X_4	X_5	$X_6 = X_1X_2X_3X_4$	$X_7 = X_1X_2X_4X_5$
1	-	-	-	-	-	+	+
2	+	-	-	-	-	-	-
3	-	+	-	-	-	-	-
4	+	+	-	-	-	+	+
5	-	-	+	-	-	-	+
6	+	-	+	-	-	+	-
7	-	+	+	-	-	+	-
8	+	+	+	-	-	-	+
9	-	-	-	+	-	-	-
10	+	-	-	+	-	+	+
11	-	+	-	+	-	+	+
12	+	+	-	+	-	-	-
13	-	-	+	+	-	+	-
14	+	-	+	+	-	-	+
15	-	+	+	+	-	-	+
16	+	+	+	+	-	+	-
17	-	-	-	-	+	+	-
18	+	-	-	-	+	-	+
19	-	+	-	-	+	-	+
20	+	+	-	-	+	+	-
21	-	-	+	-	+	-	-
22	+	-	+	-	+	+	+
23	-	+	+	-	+	+	+
24	+	+	+	-	+	-	-
25	-	-	-	+	+	-	+
26	+	-	-	+	+	+	-
27	-	+	-	+	+	+	-
28	+	+	-	+	+	-	+
29	-	-	+	+	+	+	+
30	+	-	+	+	+	-	-
31	-	+	+	+	+	-	-
32	+	+	+	+	+	+	+

Table A.1
Contrasts ("+": similar, "-": dissimilar)

$$X_1I_1 = X_1^2X_2X_3X_4X_6 = X_2X_3X_4X_6$$
$$X_1I_2 = X_1^2X_2X_4X_5X_7 = X_2X_4X_5X_7$$
$$X_1I_3 = X_1X_3X_5X_6X_7$$

Consequently the effect of X_1 is confounded with the effect of the interaction of factors X_2, X_3, X_4, X_6 and confounded with the effect of the interaction of factors X_2, X_4, X_5, X_6. Assuming the effects of these four-factor interactions are negligible, the design allows the effect of factor X_1 to be determined. Proceeding in a similar fashion for factors X_2 through to X_7 and the two-factor interactions X_1X_2 through to X_6X_7 we find that only one of the two-factor interactions is confounded with another two-factor interaction: X_3X_5 is aliased to X_6X_7, which is expected from the choice of generators and implicit in the identity equation for I_3 above.

An alternative 3-generator, 1/8 fraction, 2^{7-3} resolution IV design given in [56] with 16 runs was rejected due to the aliasing structure. With the generators $X_5 = X_1X_2X_3$, $X_6 = X_2X_3X_4$, $X_7 = X_1X_3X_4$ the identity equations were derived as follows:

$$
\begin{aligned}
I_1 &= X_1X_2X_3X_5 \times X_2X_3X_4X_6 = X_1X_4X_5X_6 \\
I_2 &= X_1X_2X_3X_5 \times X_1X_3X_4X_7 = X_2X_4X_5X_7 \\
I_3 &= X_2X_3X_4X_6 \times X_1X_3X_4X_7 = X_1X_2X_6X_7 \\
I_4 &= X_1X_2X_3X_5 \times X_2X_3X_4X_6 \times X_1X_3X_4X_7 = X_3X_5X_6X_7
\end{aligned}
$$

Resulting in the following alias structure with multiple confounds:

$$
\begin{aligned}
X_1X_2 &= X_3X_5 = X_6X_7 \\
X_1X_3 &= X_4X_7 \\
X_1X_4 &= X_5X_6 \\
X_1X_5 &= X_2X_3 = X_4X_6 \\
X_1X_6 &= X_4X_5 = X_2X_7 \\
X_1X_7 &= X_3X_4 = X_2X_6 \\
&\dots
\end{aligned}
$$

While the main-effects are uniquely resolved, the alias structure shows too many 2-factor confounds to be useful for exploratory research. The quarter fraction design thus presents the best compromise between the requirements to examine as many possible factors in this exploratory research and the resource requirements of the study.

A.2 Blocking

To avoid subject fatigue, no subject can be subjected to all 32 conditions. We must therefore employ a balanced blocking design, with either 4 blocks of size 8 or 8 blocks of size 4 for one replication of the design. One consideration is the number of generators required for the blocking, as these will be confounded with all other contrasts. 8 blocks require 3 generators, while 4 blocks require only 2 generators. Another consideration is the economy of the design with respect to the required number of subjects. As subjects need to be experienced data integration professionals, the sample frame is limited, so that a design with 4 blocks of 8 is advantageous, and still represents a good tradeoff between subject fatigue and the ability to identify the source of effects. For the blocking generators, we re-use the X_6 and X_7 generators in the fractional design: $B_1 = X_6 = X_1X_2X_3X_4$ and $B_2 = X_7 = X_1X_2X_4X_5$.

To be able to separate the block (subject) effects from other effects requires a replication of the design, so that a total of eight subjects are needed. With this replication, the design becomes a mixed within-subject and between-subject design (also called "split-plot" or "repeated measures" design), with X_6 and X_7 as between-subject factors, and $X_1 \ldots X_5$ as within-subject factors.

B Experimental Stimulus

You are part of a database integration project. As part of a business merger, two relational databases need to be integrated into a federated database. Each of the two databases of two similar sized companies is used by its own set of OLTP (online transaction processing) software applications. Each of the two databases and associated applications concerns the same business domain, the customer management for the marketing departments.

For both databases, you have available an Entity-Relationship Diagram (logical level), as well as information about the table contents (physical level). Each entity in the Entity-Relationship Diagram corresponds directly to a table in the database. You also have some information about the software applications available, such as update behavior. Neither of the databases uses stored procedures, triggers, or user-defined functions.

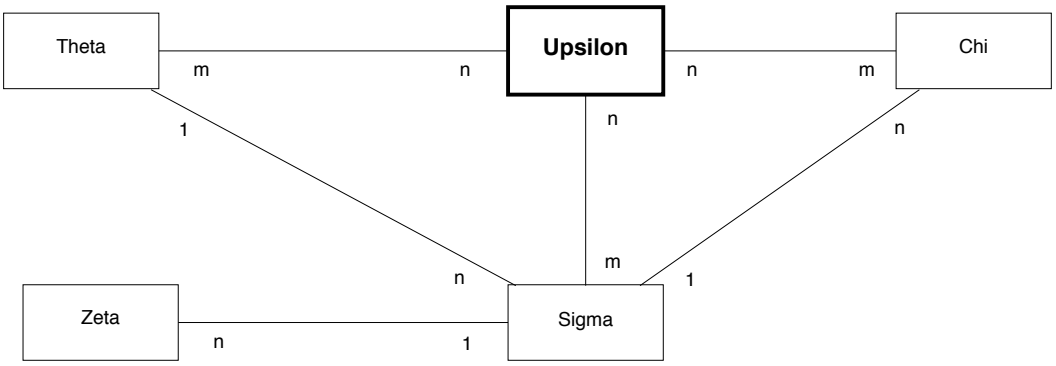
In order to integrate the two databases into a federated, virtual databases, you must decide which of the entities to match.

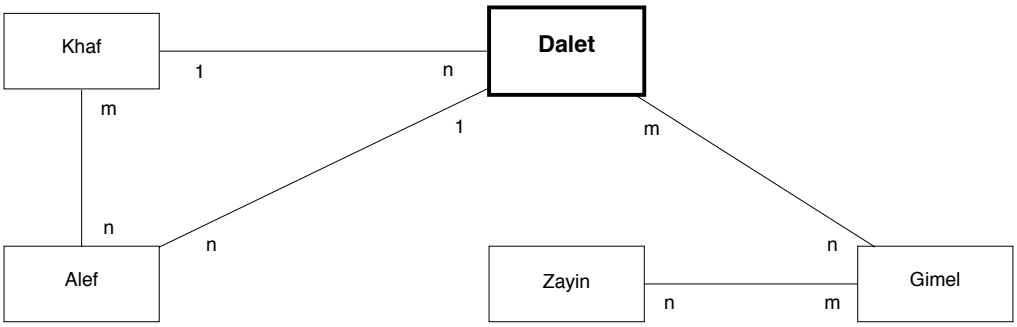
Your recommendations will be thoroughly reviewed before any integration is undertaken and you carry no responsibility for the final integration decision

made at a later stage.

Remember both databases deal with the customer management of the marketing department and no integration will be performed until your assessment has been thoroughly reviewed.

Consider all the information in the two Entity-Relationship-Diagrams on this page and the information about the corresponding database tables on the following pages. Pay attention to the elements **Upsilon** and **Dalet**.





An analysis of the *database effects of changes/updated* shows the following:

- For the first database:
  Whenever a row is deleted from the table for Upsilon, one row is deleted from Chi and one row is deleted from Sigma.
  Update frequency for Upsilon: $\approx$ 400 Updates per day.
  Insert frequency for Upsilon: $\approx$ 100 Inserts per day.
- For the second database:
  Whenever a row is deleted from the table for Dalet, one row is deleted from Gimel and one row is deleted from Khat.
  Update frequency for Dalet: $\approx$ 410 Updates per day.
  Insert frequency for Dalet: $\approx$ 110 Inserts per day.

An analysis of the *application software effects of changes/updates* shows the following:

- For the first database:

Operation on Upsilon	Software function performing the operation
Update	PRD_MNG_BASE, PRD_MNG_OPT_1, PRD_MNG_OPT_2
Insert	PRD_MNG_OPT_2, PRD_MNG_OPT_3

- For the second database:

Operation on Dalet	Software function performing the operation
Update	PRD_MNG_BASE, PRD_MNG_OPT_1,
Insert	PRD_MNG_OPT_2, PRD_MNG_OPT_3

Analysis of the two databases shows the following *attributes, data types and constraints*:

- In the first database, the table for Upsilon has the following attributes:

Attribute	Datatype	Constraints
AAN	Numeric(2)	Not null, primary key
TUD	Char(10)	
KPO	Numeric(5)	Not null

- In the second database, the table for Dalet has the following attributes:

Attribute	Datatype	Constraints
SEF	Char(5)	Primary key, check(length(SEF > 1))
ULN	Integer	Not null
IMB	Char(30)	

An analysis of the *information stored in the database tables* shows the following:

- In the first database, the table for Upsilon contains the following typical data:

AAN	TUD	KPO
13	101_X	50960
14	178_X	55990
15	192_Y	68950
16	245_X	78450
17	300_X	89990
18	290_Y	85790

- In the second database, the table for Dalet contains the following typical data:

SEF	ULN	IMB
13	101	50950_G
14	178	55990_G
15	192	68950_N
16	245	78450_G
17	300	89990_G
18	290	85790_N

An analysis of *aggregate information* in the two databases shows the following:

- In the first database:
  Total number of Rows in Upsilon (size): 9860

  Average for AAN: 18.3 Standard Deviation for AAN: 4.7

  Average for TUD: 5.0 Standard Deviation for TUD: 0

  Average for KPO: 72402 Standard Deviation for KPO: 8477

  Note: Average and standard deviation for character fields are calculated based on the length of the data.
- In the second database:
  Total number of Rows in Dalet (size): 3066

  Average for SEF: 2 Standard Deviation for SEF: 0

  Average for ULN: 198.2 Standard Deviation for ULN: 74.6

  Average for IMB: 7 Standard Deviation for IMB: 0

  Note: Average and standard deviation for character fields are calculated based on the length of the data.

C Detailed Results

This appendix contains two tables showing the detailed statistics from the pilot-test (Section 4.3) and the final experiment (Section 6).

The pilot-test results on the seven questions measuring perceived similarity were analyzed for uni-dimensionality using both ML extraction of a single factor and SEM CFA (covariance matrix based on pairwise complete observations). The results are shown in the first two columns of Table C.1. We noticed that the first item did not highly correlate with the remaining six and this is reflected in the low factor loadings/regression coefficients. We therefore re-specified the model and excluded this item. The result shows a much improved model fit for both the ML and CFA in the second two columns of Table C.1.

	7-item scale		6-item scale	
	ML	CFA	ML	CFA
Q1	.313	.359		
Q2	.735	.791	.737	.788
Q3	.843	1.00 [2]	.847	1.00 [3]
Q4	.631	.604	.638	.607
Q5	.750	.788	.750	.784
Q6	.552	.575	.539	.559
Q7	.577	.592	.567	.579
Cronbach α	.8164		.8417	
χ^2	10.6	13.20	6.44	7.96
df	14		9	
p	.717	.511	.695	.538
Prop Var Exp	.421		.474	
Coeff Det		.8689		.8684
GFI		.9386		.9588
AGFI		.8772		.9038
RMSEA		0		0
NFI		.9088		.9407
TL NNFI		1.0097		1.0145
Bentler CFI		1		1
SRMR		.0576		.0418

Table C.1
Pilot-test results

Loadings for the remaining six items remained stable, further supporting the validity of the measurement.

While the pilot-test indicated a better fit for the six-item scale for perceived similarity, the final data collection still included all seven items (Table C.2). All correlations were highly significant, however those of the first item were lower. These results were again analyzed for uni-dimensionality using both ML extraction of a single factor and SEM CFA (covariance matrix based on pairwise complete observations). The results are shown in the first two columns of Table C.3. Confirming the pilot-test results, the first item did not highly correlate with the remaining six and we therefore re-specified the model and excluded this item. The result shows a much improved model fit for both the ML and CFA in the second two columns of Table C.3. All further analyses were conducted with regression scores based on the ML extraction of one factor from the six-item scale. In contrast to using the question average, factor scores take into account the uniquenesses of the items, and leave only the common factor contribution for further analysis. It also simplifies further analysis by centering the scores on zero.

	Q2	Q3	Q4	Q5	Q6	Q7
Q1	0.6029***	0.4232***	0.3995**	0.4429***	0.4427***	0.3928**
Q2		0.6892***	0.7004***	0.5750***	0.6556***	0.6890***
Q3			0.8098***	0.7182***	0.6957***	0.8403***
Q4				0.7556***	0.7823***	0.8254***
Q5					0.6581***	0.7325***
Q6						0.7587***

Table C.2
Correlation matrix for perceived similarity scale (Sig: * = 0.05, ** = 0.01, ***=0.001)

	7-item scale		6-item scale	
	ML	CFA	ML	CFA
Q1	.490	.531		
Q2	.770	.850	.761	.839
Q3	.891	1.00 [4]	.892	1.00 [5]
Q4	.914	.967	.916	.970
Q5	.806	.883	.804	.881
Q6	.829	.933	.827	.931
Q7	.913	.955	.916	.958
Cronbach α	.9275		.9402	
χ^2	21.12	22.48	5.38	5.69
df	14		9	
p	.0987	.0692	.800	.770
Prop Var Exp	.662		.731	
Coeff Det		.9519		.9517
GFI		.9174		.9713
AGFI		.8348		.9330
RMSEA		.0981		0
NFI		.9394		.9832
TL NNFI		.9636		1.0171
Bentler CFI		.9756		1
SRMR		.0487		.0167

Table C.3
Final-test results