Rationality of Cross-System Data Duplication: A Case Study

Wiebe Hordijk and Roel Wieringa

University of Twente, The Netherlands
{hordijkwtb,roelw}@cs.utwente.nl

Abstract. Duplication of data across systems in an organization is a problem because it wastes effort and leads to inconsistencies. Researchers have proposed several technical solutions but duplication still occurs in practice. In this paper we report on a case study of how and why duplication occurs in a large organization, and discuss generalizable lessons learned from this. Our case study research questions are why data gets duplicated, what the size of the negative effects of duplication is, and why existing solutions are not used. We frame our findings in terms of design rationale and explain them by providing a causal model. Our findings suggest that next to technological factors, organizational and project factors have a large effect on duplication. We discuss the implications of our findings for technical solutions in general.

Keywords: Data duplication, design rationale, field study.

1 Introduction

Data duplication is the phenomenon that two or more systems store and maintain representations of the same real-world fact. For example, two systems may store a list of all the countries in the world and their names; one system may have them in a database table and the other in a file, or the character sets in which they are stored may differ, or the database schemas. All these cases are examples of data duplication, because different data represent the same real-world facts and are separately maintained (not replicated). Data duplication is an important problem in practice, because it leads to wasted effort to keep the duplicated data consistent, errors due to inconsistencies in data and effort to correct those errors.

Current published solutions to data duplication view it as a technical problem, which can be solved for example by ensuring connectivity between systems over a network, by transforming data to a different syntax, or by matching data based on formally defined semantics [10]. Many solutions focus on automating the task of finding out which records in one or more information systems represent the same real-world facts. For example, some approaches match records based on the similarity of their contents [1,3,16], or on the similarity of their relations with other records [2]. A recent proposal uses Bayesian networks to represent evidence for possible object matches [6]. Some approaches try to match definitions in ontologies [8,14] and others use schema matching techniques [11]. Data

B. Pernici (Ed.): CAiSE 2010, LNCS 6051, pp. 68–82, 2010.

duplication is often cited as one of the major problems ERP systems can solve, but integration with existing systems poses its own risks [13].

Despite this plethora of technical solutions, data duplication and its associated problems still exist in practice. One explanation for this is that different databases can only be merged when they share an institutional world, for example because they must interoperate in a value chain or because they report to a common regulator [4]. In all other cases, data will remain duplicated. However, this does not explain why data is duplicated within one (large) organization, i.e. one institutional world. And unless we have such an understanding, we cannot predict with reasonable certainty what technical solutions would be effective.

In order to understand the problem of data duplication better, in section 4 of this paper we will analyze a case study in-depth. This will yield an improved understanding of the causes of duplication and the size of its effects. We describe our findings in the form of Reusable Rationale Blocks, which are tables that summarize the arguments for and against different design options. We then generalize our findings in the form of a cause/effect graph showing hypotheses about the causes of data duplication, and finally speculate about organizational and technical solutions to avoid duplication and to mitigate its negative effects.

2 Case Context

The organization in which we have performed this case study is a large not-for-profit organization in The Netherlands with a strong international presence. It employs about 3000 workers in The Netherlands and 2000 abroad.

During the study period the first author worked as a consultant for the organization. He was involved in application portfolio management, which deals with improvements in the information systems in place in the organization. The results of this study in a different form were used to shape these improvements.

3 Research Design

In our case study we try to answer the following research questions:

Q1. Why does data get duplicated?
Q2. (a) What are the negative effects of this, and (b) how costly are they?
Q3. (a) How can the negative effects of duplication be avoided, (b) what role can existing technical solutions play in this, and (c) why are available technical solutions not used in our case?

We started by mapping hypothesized causes and effects of cross-system data duplication in the form of a cause/effect graph based on literature, anecdotes by stakeholders in the organization, and our own experience. This graph shows what organizational and technical factors lead to duplicated maintenance of data, and problems this duplication causes (the final version is shown in figure 1). These cause-effect relations were hypotheses in our research, i.e. we expected them to be confirmed or falsified in this particular case.

In the second phase we quantified the problem by counting inconsistencies in a number of data sets in several systems, and by estimating the effort involved in various activities needed due to duplication of data, such as duplicate entry, checks and corrections. In some cases the effort was recorded, in others personnel could give an estimate. This quantification underscores the importance of mitigating data duplication.

The third phase is more exploratory and involves searching for the mechanisms through which duplication is brought about. The unit of investigation was the individual decision, made in a project, to integrate a system under design with another system using particular technology, or not to integrate it. We assume that such decisions are made under bounded rationality, that is, decision makers make rational decisions given the limited information and time they have available [12]. This means that we needed to find the argumentation actually used in past projects. To be able to compare those arguments, we will present them in the form of reusable rationale blocks (RRBs). RRBs [5] are a simple technique to present advantages and disadvantages of options to solve a problem in a generalized table format so that they can be reused across similar problems. We have derived the argumentations from project documentation and interviews with the original stakeholders.

From the RRBs of individual projects we can see that some arguments are more important than others, depending on the context of the project. These context factors should appear as causes in the cause/effect graph created in the first phase. The arguments as represented in the RRBs are the mechanisms by which those context factors increase duplication. That way the rationale blocks both validate and explain the cause/effect graph. From the rationale blocks we will derive an improved cause/effect graph in the fourth phase.

In the fifth phase we analyze the generalizability of the results from the previous phase. Though our results come from a single organization, the research design is a multiple case design with multiple embedded units of research [17] and we can make a case for analytical generalization, also called "accumulation of knowledge" by realist methodologists [9]. In our case we intend to accumulate design-oriented knowledge that is true in sufficiently many cases without claiming to be true always in all cases. Practical knowledge often accumulates at this intermediate level of generalization [15].

The sixth phase consists of deriving practical solutions from our findings for organizations to minimize cross-system data duplication and to mitigate its negative effects. We will also discuss in which contexts the technical solutions proposed so far could contribute to the mitigation of the problem.

4 Results

4.1 Initial Cause/Effect Graph

A workshop was held in July 2009 with 10 representatives from the organizations IT department, not for the purpose of this research but to identify problems and solutions concerning application integration. The workshop was organized by the

first author together with another consultant, and attended by system maintainers, programmers, designers, an enterprise architect and a senior manager from the organization. From the results of the workshop the first author has drawn an initial cause/effect graph.

4.2 Identifying Duplicate Data

We have created a matrix matching systems against the data stored in those systems, and from it we have identified the following data objects which are used in at least three systems. We have only counted systems where the data were maintained by users, not systems which merely store a copy of data which they automatically receive from another system.

- Countries, including their names, sometimes in different languages, and country codes such as from ISO 3166 and national standards.
- Nationalities occur in some of the systems storing countries.
- Employees, sometimes with specific information such as phone numbers, sometimes only with their account name for authorization purposes.
- Organizational units
- Foreign offices, a generic name for operations of the organization in other countries. There are about 150 of them. Some systems store only a subset because their business processes apply to specific types of offices.
- Contacts, a generic name for all kinds of external parties, including persons and other organizations that the organization deals with. Many systems have their own subset of contacts, but there is overlap, e.g. when a contact occurs in the financial system, a CRM system and someones email contacts directory.

4.3 Quantification of the Problem

We have counted the number of inconsistencies among systems concerning several of the data sets listed above by manually comparing database dumps from the systems and then estimated the wasted effort caused by this. We give the results for inconsistencies between countries tables and we give estimates of the effort wasted by duplication in general.

Inconsistencies in Country Lists. We compared the countries tables with the ISO 3166[1] standard, for which the Dutch Language Union[2] provides official Dutch country names. In table 1 we list the inconsistencies per system and per type as percentage of the total number of countries. Inconsistencies are classified as 'unmatched' countries, where a country is missing from either the system or the standard, or 'spelling' where the spelling of the name differs. We have also estimated the size of the user groups and the usage frequency. We can make some observations on this data.

[1] `http://www.iso.org/iso/country_codes.htm`
[2] `http://taalunieversum.org/taalunie/`

Table 1. Inconsistencies in countries tables per system as percentage of total number of countries

System	Introduction	Users	Frequency	Unmatched	Spelling	Total
System 1	1-1-1997	5000	daily	4%	8%	12%
System 2	1-1-1998	20	daily	8%	12%	20%
System 3	1-4-2001	20	daily	7%	6%	13%
System 4	1-8-2001	500	5x per year	50%	5%	55%
System 5	1-8-2002	20	daily	14%	9%	23%
System 6	1-1-2003	500	daily	8%	26%	33%
System 7	1-10-2003	5000	daily	4%	3%	7%
System 8	1-1-2004	10	ad hoc	8%	13%	21%
System 9	1-10-2006	30	daily	11%	19%	30%
System 10	1-1-2007	5000	ad hoc	7%	6%	13%
System 11	1-1-2007	500	daily	14%	14%	28%
System 12	1-7-2007	500	1x per year	9%	23%	32%
System 13	1-6-2008	10	ad hoc	10%	21%	31%

Some systems have many unmatched countries. This can be explained because these systems only need specific subsets of countries, e.g. only those in which the organization has offices, or because countries have ceased to exist, such as Yugoslavia or the USSR, but have not been removed from a system. This may not be a problem for users of a system itself, but does pose problems when trying to integrate data from multiple systems, such as for reporting.

Spelling differences are remarkably frequent. Spelling issues arise often with respect to geographical names, but when analyzing the data we found that many of these inconsistencies were simple typing errors.

There is no relation between the age of a system and the number of inconsistencies, though we will see some evidence in subcases 3 and 4 in paragraph 4.4 that systems built on older technological platforms are harder to integrate. We do observe that generally systems with larger user groups have fewer inconsistencies than less-used systems. This can probably be explained because much-used systems receive more corrections at the request of their users.

Effort due to Duplication. In System 4, data about foreign offices are periodically aggregated from several other systems for financial and policy evaluation and prediction. These data are exported from other systems as MS Excel files, after which some manual polishing and formatting is done and the files are imported in System 4. Examples of this 'polishing' are removing negative numbers and checking total amounts after importing files. This process happens 5 times per year and takes an estimated total of 16 person-hours of all workers involved each time. This estimate was given by the application maintenance staff.

System 12 is a reporting system in which data from a large number of other systems are brought together to create management overviews of foreign operations.

Table 2. Integration effort per system

System	Integration effort
System 1	Inconsistencies in data cause users to request corrections through service desk
System 4	File export/import, communication between maintainers, data checks, manual corrections (estimate 16 hrs per time, 5 times per year)
System 5	Effort is hidden in large effort for periodical data consolidation
System 6	Changes in countries via service desk
System 9	Changes in countries via service desk. Manual corrections in output by users.
System 11	Manual maintenance of translation tables for systems to exchange data with.
System 12	Reporting takes 4 person-months per year.

Partly because of differences in reference data (such as countries) in the source systems, creating these reports takes a lot of manual effort. The recurring effort for producing these reports is 4 person-months.

Table 2 lists qualitative and some quantitative data about effort spent on manual integration of data between systems. The total waste of effort adds up to several person-months per year.

4.4 Identifying Mechanisms That Cause Duplication

If we want to build a system without duplicating the effort to maintain the data, then the first thing we need to do is find out whether there is a source system to get the data from. The data in the source system must be of sufficient quality to serve the processes of the target system. Quality attributes of the data on which integration options have influence are (adapted from [7]):

- Accuracy: the degree to which data correctly reflect an external truth.
- Completeness: the degree to which data reflect an entire external truth.
- Timeliness: the speed with which data are adapted to changes in an external truth.

When a source system with suitable data quality is found, there are some options to integrate the data between the source and target system. Each of these possibilities has advantages and disadvantages which the project must weigh against the success criteria that the stakeholders apply to this project. This can make it logical for a project to deliver an isolated system, that duplicates data stored elsewhere too, even if that is disadvantageous for the organization as a whole.

- Manual re-typing of the data. This is flexible and no investment is needed but takes a lot of effort and usually leads to many typing errors.
- Manually exporting and importing files. Also very flexible and takes a small project investment but still costs effort, and errors still occur, especially when manual "improvements" to the data are necessary.
- Automatically exchanging files. Takes a slightly bigger investment. Errors still occur, e.g. due to incomplete files.

Table 3. RRB showing the general pros and cons of integration options. ++ is very good value and - - a very bad value for the stakeholder, respectively.

Stake-holder	Quality attribute	Impor-tance	Manual typing	Manual files	Autom. files	DB link	Messag-ing
Project; Owner	Project investment (+ is lower)	+	++	+	- -	-	- -
Project	Project risk (+ is lower)	+	++	+	- -	-	- -
Owner; Maintenance team	Independence of other systems	+	++	+	-	- -	-
Owner; User	Business process flexibility	+	++	+	- -	- -	-
User; Manager	Data integration effort (+ is lower)	++	- -	-	+	+	+
User	Data accuracy	++	- -	-	+	+	+
User	Data timeliness	++	- -	- -	-	+	+
Maintenance team	Maintainability	+	+	-	-	+	++

- Database link to the source system. The investment is often smaller than with file export/import but the solution is less flexible because the source system cannot change its data structure without breaking the target system. The data are always up to date but the target system will depend for its availability on the source system and the network.
- Messaging, e.g. using web services. This option takes the largest investment but it is more flexible than a database link and more reliable and flexible than file exchange.

Together with professionals from the organization we have identified general quality attributes for these options. This general knowledge was codified in the form of a RRB in Table 3. It should be read as follows: all else being equal, the data integration effort will be worst (= highest) when manual typing is chosen and best when one of the automated options is chosen; data integration effort will usually matter more to stakeholders than the initial investment. We do not claim that the RRB of Table 3 is true in all cases but we do claim that this analysis method can be used in similar cases too, and will often contain very similar decision rationales.

We have identified four decisions in past projects to integrate systems in a certain way by interviewing the original stakeholders and reviewing project documentation. We have investigated how the context factors of a specific project shape the arguments that feed into the decision and lead to different outcomes. These arguments, represented below by means of RBBs, will be used to derive and corroborate hypotheses about causes and effects of data duplication. We aggregate the hypotheses into a causal effect graph in Figure 1 and Table 6.

We have noticed in this phase that most project decisions are not documented with their associated rationale (and sometimes not at all). This makes it essential

for case study research to have access to original stakeholders, which has driven our choice of cases.

Subcase 1. System 11: Countries. When system 11 was under development, architects have actively looked for a suitable source system for countries and nationalities. Documentation describes why systems 5 and 6 cannot deliver data with the required quality. Other source systems have not been considered, and architects involved in the process indicate that it was unknown at the time that other systems also stored country tables. The decision was made to give System 11 its own countries table. This case leads to hypothesis *H1: lack of knowledge about the architectural landscape of the organization is a contributing factor to data duplication by increasing the cost of search for potential source systems.*

When systems 5 and 6 turned out not to have the desired data quality, the project first tried to request from the respective owners of those systems to improve the data and then share it. The owners of both systems responded in a tactful way that they did not see this as their task. This teaches us that for systems to integrate their data an organization first needs to establish the organizational responsibilities and capabilities to maintain and provide data with the right quality. We postulate two hypotheses. *H2: poor or unknown data quality in potential source systems leads to data duplication. H3: Unwillingness to establish organizational dependencies leads to data duplication.*

Subcase 2. System 11: Currencies. At the same time when designing system 11, the system needed data about which currencies can be used in countries in the world and spot rates of those currencies against the euro. The currencies are updated about as often as countries themselves, but spot rates change daily. System 6 is a natural source for these data because it is an ERP system used for financial administration throughout the organization. It can be trusted to have accurate data about currencies and rates. This is consistent with hypothesis H2.

Table 4 shows the advantages and disadvantages of integration options between systems 6 and 11, adapted from Table 3 to the specific context factors of this project. We use the table to detect how variations in context factors for individual projects determine which option is best in individual decisions.

System 11 is a relatively new system built on a modern development platform and uses messaging between its own components. That makes the project investment and the project risk for the option 'messaging' lower. The same holds for 'automated file transfer', with which the maintenance team of system 6 has broad experience. We postulate hypotheses *H4: availability of integration infrastructure decreases the chance of data duplication by decreasing the relative cost of integration.*

The development of system 11 was a very large project of which this particular integration formed only a small part. This reduces the importance of the project investment. *H5: High pressure on a project increases the chance of data duplication by increasing the importance of project risk and project investment.*

For the business process of system 11 it is important that currencies and rates are received correctly each day. This increases the relative importance of

Table 4. Pros and cons of integration options for currencies in System 11; differences with Table 3 in brackets

Stake-holder	Quality attribute	Impor-tance	Manual typing	Manual files	Autom. files	DB link	Messag-ing
Project; Owner	Project investment (+ is lower)	(-)	++	+	(-)	-	(-)
Project	Project risk (+ is lower)	+	++	+	(-)	-	(-)
Owner; Maintenance team	Independence of other systems	+	++	+	-	- -	-
Owner; User	Business process flexibility	+	++	+	- -	- -	-
User; Manager	Data integration effort (+ is lower)	(+++)	- -	-	+	+	+
User	Data accuracy	(+++)	- -	-	+	+	+
User	Data timeliness	(+++)	- -	- -	(+)	+	+
Maintenance team	Maintainability	+	+	-	-	+	++

accuracy and timeliness. Because the rates only change daily and automated files are suitable for daily transfers, the automated files option gets a + for timeliness. Because of the large amount of data to be transferred each day, data integration effort is relatively important. *H6: high-volume data is less susceptible for duplication than low-volume data, because the data integration effort is higher.*

In Table 4 we can see that the three automated options are still viable. Database link was not chosen because it leads to tight coupling between the systems. Messaging was favored over automated file transfer because it was thought to lead to more maintainable systems.

Subcase 3. System 4: Countries and financial data. System 4 is used to make budgets and to evaluate the financial performance of the organization as a whole. It receives financial information from system 6, the ERP system, and the countries list in system 6 is good enough for system 4, so system 6 is a viable source system. The advantages and disadvantages of the options to connect these systems are listed in Table 5.

System 4 is an old system built in MS Access, which does not offer standard messaging technology, just like the ERP system 6. Use of messaging is only possible with a large investment in technology and would incur considerable risk to a project. This corroborates hypothesis H4: availability of integration infrastructure decreases duplication.

There are plans to rebuild system 4 with .NET technology to a web-based system. This means that any investment in system 4 in the short term should be profitable in a short time. This increases the relative importance of the quality attribute 'Project investment'.

The data collected in system 4 and its business rules change slightly from year to year. These changes typically are agreed upon shortly before a new version

Table 5. Pros and cons of integration options for system 4; differences with Table 3 in brackets

Stake-holder	Quality attribute	Impor-tance	Manual typing	Manual files	Autom. files	DB link	Messag-ing
Project; Owner	Project investment (+ is lower)	(++)	++	+	- -	-	(- - -)
Project	Project risk (+ is lower)	+	++	+	- -	-	(- - -)
Owner; Maintenance team	Independence of other systems	+	++	+	-	- -	-
Owner; User	Business process flexibility	(++)	++	+	- -	- -	-
User; Manager	Data integration effort (+ is lower)	(+)	- -	(-/+)	+	+	+
User	Data accuracy	++	- -	(-/+)	+	+	+
User	Data timeliness	(+)	- -	- -	-	+	+
Maintenance team	Maintainability	+	+	-	-	+	++

of the system goes live. This sometimes makes it necessary to manually change the contents of files transferred from system 6 to system 4. This increases the relative importance of the quality attribute 'Business process flexibility'. This leads to *H7: required flexibility in business processes can lead to data duplication.*

Data integration between these systems is performed by employees who have been doing this for almost 10 years now. They know exactly what to do and for what kinds of errors to check. For the systems architecture this means that the option of manual file transfer scores better on the quality attributes 'Data integration effort' and 'Data accuracy'. We propose *H8: expertise in maintaining duplicated data keeps duplication in place by minimizing its drawbacks.*

Data timeliness is less important for this integration because the whole process happens only 5 times per year. Under these specific circumstances, manual file transfer has the optimal balance between project investment and integration effort. This motivates *H9: frequently updated data is less likely to be duplicated than data which is less frequently updated.*

Subcase 4. System 12: Reporting data. System 12 is a system in which large amounts of data about the organizations foreign offices are aggregated into management information. Before this system was developed there was a lot of uncertainty about how and in what formats source systems would be able to deliver their data. For those reasons, business process flexibility and independence of other systems were very important. This corroborates hypotheses H2: poor or unknown data quality increases duplication, and H7: required flexibility in business processes can lead to data duplication.

The amount of data was not too large to make manual checks and corrections unfeasible. The reports from the system were only needed once per year. This decreases the relative importance of Data integration effort, Data timeliness and

Maintainability. This corroborates H6: high-volume data is duplicated less often, and H9: more frequently updated data is duplicated less often.

The resulting rationale table for system 12 is very much like that for system 4 in Table 5, and for reasons of space we omit the specific table for system 12. For the integration of data into system 12, manual file transfer was chosen for much the same reasons as for system 4.

4.5 Lessons Learned

Based on the hypotheses described in the cases above we have adapted the cause-effect graph drawn at the start of our project. The result is shown in figure 1.

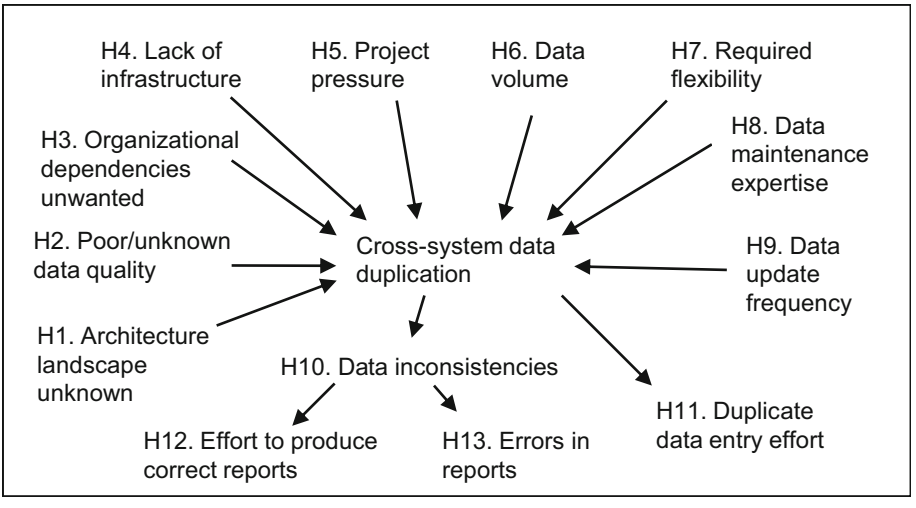

Fig. 1. Final Cause/Effect graph. Nodes are observable phenomena, marked with the hypotheses in which they play a role; arrows are positive influences

The nodes in Figure 1 are marked with the hypothesis numbers in which they play a role. In Table 6 we summarize the hypotheses with their evidence.

4.6 Generalizability

It is not possible to find in one case study arguments for generalization to all other cases. But it is possible to indicate which claims could be generalized to an interesting set of other cases. We claim that our results are generalizable to, and therefore reusable in, situations where the same mechanisms are in place as in our case study.

4.7 Solutions

Solutions to the problems around data duplication consist of a mix of organizational, technical and project management elements, just like the problems

Table 6. Hypotheses and their evidence

Hypothesis	**Evidence**
H1: lack of knowledge about the architectural landscape of the organization is a contributing factor to data duplication by increasing the cost of search for potential source systems	Observation in Subcase 1. In the other subcases, source systems were found, but the search effort spent is unknown. Intuitive and matches anecdotal evidence from other organizations.
H2: poor or unknown data quality in potential source systems leads to data duplication	In subcase 1, data was maintained for a specific business process and did not meet the quality requirements of another process. In subcase 4, quality of available data was not known at design time, which led to duplication. In subcase 2 good data quality enabled integration. Intuitive and matches anecdotal evidence from other organizations.
H3: Unwillingness to establish organizational dependencies leads to data duplication	In Subcase 1, a potential data supplier was unwilling to maintain data at a higher quality than needed for their own process. Seems organization-dependent.
H4: availability of integration infrastructure decreases duplication.	In subcase 2, messaging infrastructure was available. In subcase 3 it was not; it would have made messaging a more likely option. Intuitive and matches anecdotal evidence from other organizations.
H5: High pressure on a project increases the chance of data duplication by increasing the importance of project risk and project investment	Observed only directly in subcase 2. In other subcases we can see that investments for improvement are not made until other major changes to systems are necessary. Seems organization-dependent.
H6: High-volume data is less susceptible for duplication than low-volume data, because the data integration effort is higher	Observed in subcase 2 (high volume) and subcase 4 (also high, but outweighed by other context factors). Intuitive and matches anecdotal evidence from other organizations.
H7: Required flexibility in business processes leads to data duplication	In subcases 3 and 4, required flexibility led to duplication. Seems organization-dependent.
H8: expertise in maintaining duplicated data keeps duplication in place by minimizing its drawbacks	Observed in subcase 3. Seems organization-dependent.
H9: frequently updated data is less likely to be duplicated than data which is less frequently updated	Observed in subcases 3 and 4 (low frequency) and subcase 2 (high frequency). Intuitive and matches anecdotal evidence from other organizations.
H10: Data duplication leads to inconsistencies between data sets	Observed and quantified for countries and offices in paragraph 4.3. Intuitive and matches anecdotal evidence from other organizations.
H11: Duplication costs extra effort to maintain data in multiple systems	Observed and quantified in paragraph 4.3. Intuitive and matches anecdotal evidence from other organizations.
H12: Due to inconsistencies between data, making reports with data from multiple systems takes more effort	Observed and quantified in paragraph 4.3. Intuitive and matches anecdotal evidence from other organizations.
H13: Inconsistencies between data sets cause errors in reports	Anecdotal evidence. Intuitive and matches anecdotal evidence from other organizations.

themselves do. In this paragraph we speculate about solutions and their merits. Since this is not the main topic of this research effort, the solutions are presented only briefly.

To combat the lack of information about the application landscape we need a well-documented architecture landscape (solution S1), including information about what data is stored in which system. This documentation should be kept up to date and be accessible to projects.

The problem of systems only catering for their own business processes is hard to beat. One solution could be to establish an organizational entity that is responsible for maintaining data for the entire organization (S2). This has been done at several organizations where properly maintaining large quantities of data is important. This opens the extra opportunity to supply such data to other parties for added benefit. Another solution would be to add the responsibility for maintaining a particular data set to a department, along with the capabilities (people, budget) to fulfill that responsibility (S3). This is easier to do in some organizations than in others.

The relative importance of long term, organization-wide quality attributes should be increased at the cost of short-term, project-only interests. One possible way to achieve this is by adding an architect to the project team who gets paid by the IT organization instead of the system owner (S4). Another way is to use an architecture review process in which the designed system is reviewed by organization architects against criteria which reflect long term interests (S5).

Even if data is still being maintained in multiple systems, the problems that come with inconsistencies between the data sets can be minimized by agreeing on standards for the data and on procedures to check data against the standard (S6). For example, if our organization would agree on using the ISO 3166 standard for countries and all systems were up to date then integrating other data which uses countries as reference data would be much easier.

Central investments in infrastructure, e.g. for messaging, can decrease the project investment needed to provide an optimal solution which would otherwise be too expensive and too risky for a project (S7).

After all these organizational problems have been solved, the technological solutions listed in section 1 become viable. Data warehouse technology (S8) can make reporting easier, thus decreasing the reporting effort and reporting errors. A project to introduce a data warehouse in the organization is currently starting. First, however, inconsistencies in the data fed into the data warehouse from different systems must be reconciled. Entity linkage technologies (S9) and ontologies (S10) can help with that. These technologies are useful to integrate large amounts of data, but to start using them a considerable investment in skills will be needed. That means they are most useful in environments where the cost of manual integration and the risk posed by inconsistencies outweigh the investment in new technologies, that is, in situations where either the volume of the data are large, errors are very dangerous or expensive, or new technologies are easily integrated. Our organization does not have any of these characteristics. The volume of data in our organization is such that low-tech solutions are

adequate to solve the inconsistencies, and the organization has a policy of using only proven technology and not be technically innovative. This means that for our organization the risk associated with introducing new technologies outweighs the potential benefits.

Solutions S1, S2, S5, S6, S7 and S8 have been reported to the organization and are currently under review. Possible solutions for example involve changes to existing systems, to infrastructure, to procedures or to organizational structures. At the time of writing, these proposed changes are being considered in the organizations application and project portfolio management processes.

5 Discussion, Conclusions and Future Work

We have performed a series of historical case studies in a single organization. The external validity of such a research design is always questionable. The initial theory in the form of the cause/effect graph, the general rationale table and the information about individual cases were all taken from the organization under investigation. In paragraph 4.6 we have reflected on the generalizability of the results in the cause/effect graph of Figure 1.

We have provided quantitative results about the problems caused by data duplication. The data about the causes of data duplication is mostly qualitative in nature. At present this is the best we can do, and given the current state of the research we think that even such qualitative results are a contribution to the advancement of theory in this field.

Table 3 with its advantages and disadvantages of integration options is not meant to represent a general truth about relative preferences of these options: we do not claim that messaging for example is always better than database links; rather we have shown that these preferences are situation-dependent. The methodology of rationale blocks, however, can be reused as an intuitive tool for decision support, rationale documentation and as a research method to investigate the sensitivity of option preferences to context factors, as we have done. It is a pragmatic approach which stakeholders can quickly understand. Discussions in the course of our investigation immediately centered on the pros and cons of options, not about the notation technique or methodology. Rationale tables allow arguments for individual decisions to be reused in other situations. We have shown that the decision to integrate or not to integrate systems can often be explained using arguments that are rational from the standpoint of the decision maker.

Our results show that organizational, technical and project factors can cause data duplication across systems in an organization. We have suggested a set of mitigation strategies to reduce data duplication and to decrease its negative effects. These can be used by organizations to get projects to give higher priority to organization-wide long term interests and to deliver better integrated systems.

Future work should include repeating this research in other organizations and evaluation of the performance of the introduced solutions.

We thank the anonymous reviewers for their helpful suggestions for improving this paper.

References

1. Ananthakrishna, R., Chaudhuri, S., Ganti, V.: Eliminating fuzzy duplicates in data warehouses. In: VLDB 2002: Proceedings of the 28th international conference on Very Large Data Bases, pp. 586–597 (2002)
2. Bhattacharya, I., Getoor, L.: Deduplication and group detection using links. In: Workshop on Link Analysis and Group Detection (LinkKDD 2004), Seattle, WA, USA. ACM, New York (2004)
3. Cohen, W.W.: Integration of heterogeneous databases without common domains using queries based on textual similarity. In: SIGMOD 1998: Proceedings of the 1998 ACM SIGMOD international conference on Management of data, pp. 201–212 (1998)
4. Colomb, R.M., Ahmad, M.N.: Merging ontologies requires interlocking institutional worlds. Appl. Ontol. 2(1), 1–12 (2007)
5. Hordijk, W., Wieringa, R.: Reusable rationale blocks: Improving quality and efficiency of design choices. In: Dutoit, A.H., McCall, R., Mistrik, I., Paech, B. (eds.) Rationale Management in Software Engineering, pp. 353–371. Springer, Heidelberg (2006)
6. Ioannou, E., Niederee, C., Nejdl, W.: Probabilistic entity linkage for heterogeneous information spaces. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 556–570. Springer, Heidelberg (2008)
7. Jiang, L., Borgida, A., Mylopoulos, J.: Towards a compositional semantic account of data quality attributes. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 55–68. Springer, Heidelberg (2008)
8. Noy, N.F.: Semantic integration: a survey of ontology-based approaches. SIGMOD Rec. 33(4), 65–70 (2004)
9. Pawson, R., Tilley, N.: Realistic Evaluation. SAGE Publications, London (1997)
10. Pollock, J.T.: Integration's dirty little secret: It's a matter of semantics. Technical report, Modulant (2002)
11. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal 10(4), 334–350 (2001)
12. Simon, H.A.: The Sciences of the Artificial, 3rd edn. MIT Press, Cambridge (1996)
13. Sumner, M.: Risk factors in enterprise-wide/erp projects. Journal of Information Technology 15(4), 317–327 (2000)
14. Uschold, M., Gruninger, M.: Ontologies and semantics for seamless connectivity. SIGMOD Rec. 33(4), 58–64 (2004)
15. Wieringa, R.J.: Design science as nested problem solving. In: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, Philadelphia, pp. 1–12 (2009)
16. Winkler, W.E.: The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Census Bureau (1999)
17. Yin, R.K.: Case study research: design and methods, 3rd edn. Applied Social Research Methods Series, vol. 5. SAGE Publications, Thousand Oaks (2003)