



Data Journey Modelling: Predicting Risk for IT Developments

DOI:
[10.1007/978-3-319-48393-1_6](https://doi.org/10.1007/978-3-319-48393-1_6)

Document Version
Accepted author manuscript

[Link to publication record in Manchester Research Explorer](#)

Citation for published version (APA):
Eleftheriou, I., Embury, S., & Brass, A. (2016). Data Journey Modelling: Predicting Risk for IT Developments. In *Proceedings of the 9th IFIP WG 8.1 Working Conference on The Practice of Enterprise Modeling (PoEM)* (Lecture Notes in Business Information Processing). https://doi.org/10.1007/978-3-319-48393-1_6

Published in:
Proceedings of the 9th IFIP WG 8.1 Working Conference on The Practice of Enterprise Modeling (PoEM)

Citing this paper
Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights
Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy
If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [<http://man.ac.uk/04Y6Bo>] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.



Data Journey Modelling: Predicting Risk for IT Developments

Iliada Eleftheriou, Suzanne M. Embury, and Andrew Brass

School of Computer Science, University of Manchester,
Oxford Road, Manchester, United Kingdom
`iliada.eleftheriou@manchester.ac.uk`

Abstract. Information sharing is a vital element of a successful business. However, technological, organisational and human challenges obstruct the effective movement of data. In this paper, we analyse a collection of case studies from healthcare describing failing information systems developments. A set of 32 failing factors were extracted showing that data movement, either between systems, people or organisations, is a key indicator of IT failure. From this examination, we derived anti-patterns for data movement in which some key differences between the source and target location of the movement caused high costs to the developments. Finally, we propose data journey modelling as a lightweight technique that captures the movement of data through complex networks of people and systems, with the aim of improve go/no-go decision making for new IT developments, based on the anti-patterns we have identified.

Key words: Information sharing, Data movement modelling, Data journey, Socio-technical challenges

1 Introduction

While software has the capability to bring many benefits to organisations, it can be a mixed blessing [5]. New software developments can be unexpectedly costly to develop and run. It may be necessary to employ new personnel or retrain existing staff. New ways of working may need to be devised, to fit with the constraints of the new software and the technical infrastructure on which it runs. One does not need to spend long looking through newspaper headlines or the Risks List digest¹ to know that new software developments can sometimes result in costs that far outweigh the value they propose to create.

Clearly, a lightweight and reliable means is needed of helping us to make good go/no go decisions regarding new software developments. Current approaches to managing risk and estimating the cost of software development are principally focused on creating detailed predictions based on substantial models of the planned development [2]. They are aimed at supporting project managers throughout the development process itself, rather than giving a low-cost indicator for use in early-stage decision making. What is needed is a lightweight

¹ www.risks.org

approach, that can be completed in the course of a small number of days, and that gives reliable predictions of the likely success of a planned IT development. And since the reasons for software failure are rarely technical in nature, the indicator must take account of social and organisational factors, as well as the technologies to be used.

We set out to design an indicator of this kind for use in large complex organisations. As a starting point, we analysed a set of 18 case studies of new software developments in the UK's National Health Service (NHS). The case studies were written by NHS staff as coursework for the "Informatics for Healthcare Systems" course unit run by the University of Manchester, in the 2013 academic session. The study authors came from a variety of roles, and the studies describe new IT developments from a broad range of NHS functions, including cancer care, ambulance service management, in-patient management, heart failure care, diabetes care, bed management and more.

A common feature of the cases where the new software was deemed to have been unsuccessful was the movement of data. Whenever data was moved to new contexts, and used for a purpose other than that for which it was originally designed, the system owners and end-users faced a host of additional challenges, be they organisational, technical, human, governance oriented or political in nature [12]. These challenges lead to unforeseen costs and sometimes dramatic reductions in the benefits expected from the new software. We therefore hypothesised that identifying the need for movement of data in a new development could provide the early warning signal for success or failure that we were looking for.

To test this hypothesis, we developed a way of modelling the movement of data through and across organisations, and of identifying the kinds of data movement that lead to high risk and cost. The technique is lightweight because it abstracts away from the details of the business processes that use the data, and focuses just on the bare movement of data between significant entities. This paper describes the process we used to define it, and the basis for the model in information we extracted from the case studies. We began by extracting from the case studies a list of the root causes of failure (Section 3). We further analysed the case studies to extract the data movement patterns involved in each case, and combined this with the failure causes to produce a set of problematic data movement patterns (Section 4). From these patterns, we extracted the minimum information that must be captured about a new development in order to identify the presence of the patterns (Section 5). We called the resulting model the "data journey" model, since it models the paths data takes through the socio-technical enterprise, from point of entry to point of use.

2 Modelling Data Movement

A plethora of modelling techniques and notations have been proposed for use during information systems design, some of which include elements of the movement of data. In this section, we survey the principal modelling techniques, to

see if any meet our requirements and can be used as the basis for modelling data journeys. We need a modelling technique that:

- allows us to model the movement of data within and between organisations,
- gives equal prominence to both social and technical factors affecting the movement of data, and
- is sufficiently lightweight to be used as a decision-making aid in the early stages of a development cycle.

A number of software design techniques allow modelling of data from a technical point of view. Data flow diagrams (DFDs) are the most directly relevant of these [3]. Unfortunately, the focus in DFDs is on fine-grained flows, between low-level processing units, making it hard to capture higher-level aspects of the enterprise that can bring cost and risks, i.e. the social factors. Similarly, the Unified Modelling Language (UML) contains several diagrams detailing movement of data, notably sequence diagrams, collaboration diagrams and use case diagrams [7]. Although the abstraction level at which these diagrams are used is in the control of the modeller, to an extent, they provide no help in singling out just those elements needed to predict costs and risks of a potential development. Also, social factors influencing information portability and introducing cost to the movement aren't part of the focus of those approaches. These models are helpful in designing the low level detailed data flows within a future development, but can't help us decide which flows may introduce costs and risks to the development.

Other techniques are able to model high-level data movement between systems and organisations. Data provenance systems, for example, log the detailed movement of individual data items through a network of systems [8]. While these logs can be a useful input to data journey modelling, they describe only the flows that are currently supported and that have actually taken place. They are not suited to modelling desired flows, and do not directly help us to see what social and organisation factors affect the flow.

Business process modelling (BPM) captures the behaviour of an organisation in terms of a set of events (something happens) and activities (work to be done) [1]. Although BPM can implicitly model flows of data between a network of systems, they typically contain much more detail than is needed for our purposes. Data journey models aim to abstract away from the nitty-gritty of specific business processes, to give the big picture of data movement.

Models that combine technical and social information, such as human, organisational, governance and ethical factors, can be found in the literature [10, 11, 6]. For example, the i* modelling framework aims to embed social understanding into system engineering models [10]. The framework models social actors (people, systems, processes, and software) and their properties, such as autonomy and intentionality. Although a powerful mechanism to understand actors of an organisation, the i* framework does not give us the information flows happening between the systems, nor any measures to identify costs.

In summary, we found no existing model that met all our requirements. The extant technical modelling methods give us a way to model the detail of a system

to be constructed, but provide no guidance to the modeller as to which parts of the system should be captured for early-stage decision making and which can be safely ignored. The socio-technical models allow us to capture some of the elements that are important in predicting cost and risk in IT developments, but need to be extended with elements that can capture the technical movement of data. We therefore set out to define a new modelling approach for data journeys, focused wholly on capturing the data movement anti-patterns we located in the case studies. In the sections that follow, we describe and justify the model we produced.

3 Data Movement And IT Failure

Data movement is crucial to the functioning of most large organisations. While a data item may first be introduced into an organisation for a single purpose, new uses for that data will typically appear over time, requiring it to be moved between systems and actors, to fulfil these new requirements. Enterprises can thus be viewed, at one level of abstraction, as networks of sub-systems that either produce, consume or merely store data, with flows between these sub-systems along which data travels.

When we plan to introduce new functionality into an enterprise, we must make sure that the data needed to support that functionality can reach the sub-system in which it will be consumed, so that value can be created from it. The costs of getting the data to its place of consumption must be worth the amount of value generated by its consumption. Moreover, new risks to the enterprise will be introduced. The enterprise must evaluate the effects on its core functions if the flow of data is prevented by some reason, or if the costs of getting the data to its place of use rise beyond the value that is produced.

We wanted to understand whether this abstraction could provide a lightweight early-warning indicator of the major costs and risks involved in introducing new functionality in an enterprise. We began by examining the collection of case studies from the NHS, first to categorise each one as a successful or a failing development, and second to understand the major root causes of failure in each case. These were relatively straightforward tasks, as the authors of the case studies were asked in each case to diagnose for themselves the causes of failure.

Of the 18 case studies, only 3 were described by their authors as having been successful. The remaining 15 were categorised as having failed to deliver the expected benefits. We extracted and organised the failure factors identified by the authors, and aggregated the results across the full set of case studies. The results are summarised in Figure 1, which lists the failure factors in order of prevalence (with those factors occurring in the most case studies appearing on the left, and those occurring in the least, on the right). A brief explanation of each factor is given in Appendix A.

Clearly, the technical challenges of data movement are implicated in many of these failure factors. Costs introduced by the need to transform data from one format to another have long been recognised, and tools to alleviate the problems

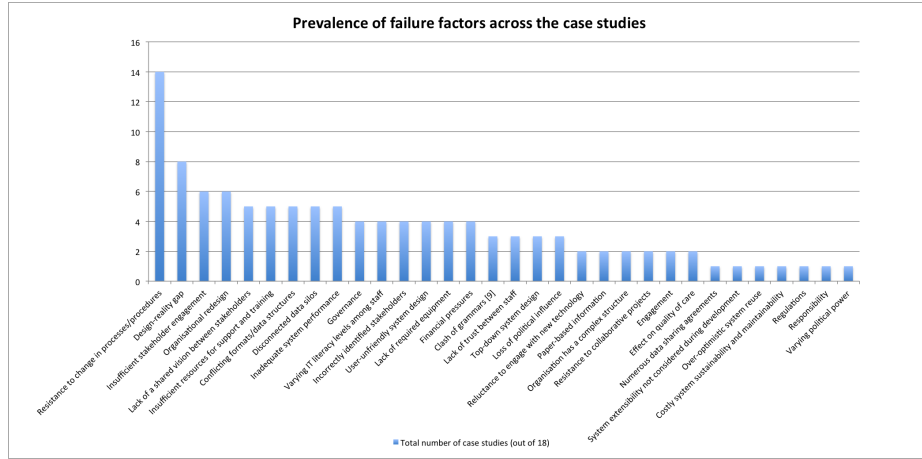


Fig. 1. Prevalence of failure factors across the case studies.

have been developed. However, from the chart, we see that the most common causes of IT failure in our case studies are related to people and their interactions. Of the 32 factors identified, less than a quarter are primarily technical in nature. Can an enterprise model focused on data movement take into account these more complex, subjective failure factors, without requiring extensive modelling?

Looking more closely, we can see that data movement is implicated in many of the non-technical failure factors, too. Many of the factors come into play *because* data is moved to allow work to be done in a different way, by different people, with different goals, or to enable entirely new forms of work to be carried out using existing data. Data moves not only through the technical infrastructure of databases and networks, but also through the human infrastructure, with its changing rules, vocabularies and assumptions. All this suggest that data movement could be a proxy for some of the non-technical risks and cost sources the study authors experienced, as well as the technical costs and challenges.

The question therefore arises as to whether we can use the presence of data movement as the backbone for our prediction model of cost and risk. If we can abstract the details of the new IT development into a sequence of new data movements that would be required to realise it, can we quickly and cheaply assess the safety of those new movements, combining both technical and social features to arrive at our assessment of the risk?

To do this, we need to understand the specific features of data movements that can indicate the presence of costs and risks. We returned to our case studies, to look for examples of data movement that were present when the IT development failed, and to generalise these into a set of data movement patterns that could become the basis for our prediction method. The results of this second stage of the analysis are described in the following section.

4 Data Movement Patterns

Having examined the case studies, we found that data movement is a key indicator of most of the IT failing factors. In this section, we propose a catalogue of data movement anti-patterns, each describing movements of data that might introduce some type of cost or risk to the development. We also give the conditions under which a pattern causes a failure, and the type of cost or risk it might impose on the organisation.

To develop the data movement patterns catalogue, we first went through the case studies and extracted any data movement or information sharing example that caused a cost or risk contributing to an IT failure. The case study authors had not been asked to provide this information explicitly in their assignment, and therefore we used our own judgement as to what data movement was involved. We then transformed those examples into a set of generic anti-patterns.

All of the case studies involved some kind of data movement and it was commonly the case that the data movement was at the heart of the part of the development that failed. Although there were many examples of movement of data between computer systems we found a richer variety of movement patterns between people, from people to systems, and vice-versa. We describe below the anti-patterns we identified from the case studies; of course, other potentially problematic data movement patterns may exist. For each pattern we give an identifying name, define the context in which it can happen, and provide the conditions that should hold for the costs to apply. Any examples given are taken directly from the case studies (but with identifying details removed).

4.1 Change of Media

Often, a change of medium is required when data is moved between a producer and a consumer. This is straightforward in the case of electronic data, which can be easily converted into report form, for document generation and printing. But the situation is more complicated when data on paper must be entered into a destination software system. Data entry is a time consuming process typically done by clerical staff, who may not have a strong understanding of the meaning of the data they are entering. Errors can easily be injected that may significantly reduce the quality of the information. We illustrate this pattern in figure 2 (a), and we define it as:

“When data moves from a source ‘S’ to a target ‘T’ of a different media (i.e. physical to electronic), then a transformation cost exists, either before or after the transportation of the data, that can lead to decreased quality at the T side.”

4.2 Context Discontinuity

Sharing data outside the immediate organisational unit can result in a number of administrative costs, such as reaching and complying with data sharing agreements, as well as complying with wider information governance requirements.

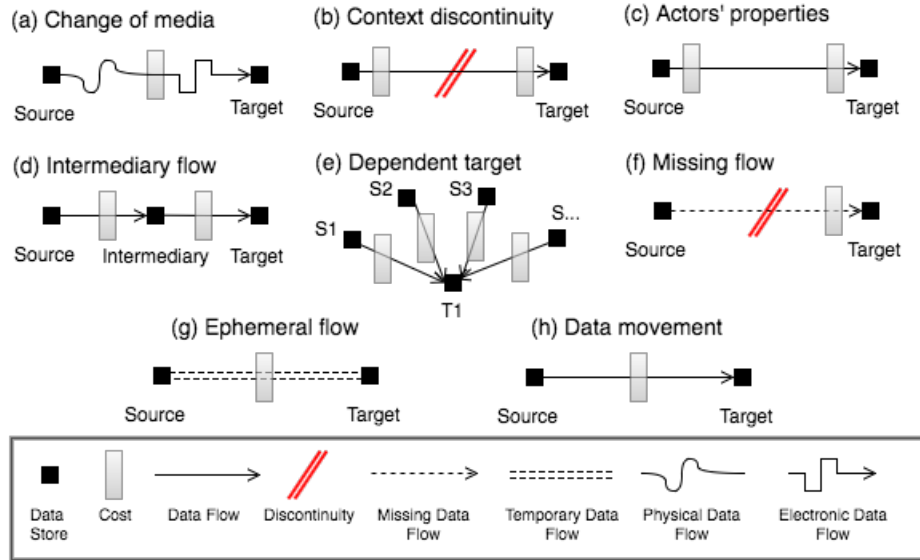


Fig. 2. Data movement anti-patterns retrieved from the case studies.

Also, a risk of staff reluctance to share ownership of data, may exist on both sides of the movement. Additionally, if the source of data belongs to a different context than the target, then there is the risk of clash of grammars (the meaning of the data moved being altered by the change of context because of a cultural, experience or other type of reason [9]), and a cost of lower data quality at the target side. For example, data entered into a system by secretarial staff can contain errors if the information requires medical knowledge/vocabulary that the staff lack. Also, if the data to be moved are in physical form (e.g. letter, cassette, X-ray film, blood sample etc.), then there are transportation costs. Generally, if there is a discontinuity in the flow of data caused by a change in context, costs will be imposed to the movement. The context discontinuity pattern is showed in figure 2 (b) and defined as:

“When data moves from a source ‘S’ to a target ‘T’ of a different context (i.e. organisation, geographical area, culture, etc.) and a discontinuity exists in the flow, then a bridging cost is imposed to either or both sides of the flow.”

4.3 Actors’ Properties

Costs can also be introduced by key heterogeneities in the properties of the consumer and producer. Differences in system requirements, business processes, governance, and regulations between producers and consumers of data create transformation costs that must be borne either at the source, or target location (or both). Integrating data from “data island” sources (sources that haven’t previously been shared up to this point) can have high costs; such sources typically

have limited external connectivity, and are tailored for use by one type of users bringing a risk of data quality problems at the target side.

“When data moves from a consumer ‘C’ to a producer ‘P’ (system or human), a difference in a property of either source or target introduces a transformation cost to the movement (figure 2-c)”.

4.4 Intermediary Flow

Intermediary systems or staff may be introduced with the aim of reducing some up-front cost (such as the use of lower-paid staff to enter data on behalf of higher-paid staff), but can actually create downstream costs in the longer term such as those caused by lower data quality or missing data.

“When data moves from a source ‘S’ to a target ‘T’ through an intermediary step, a cost is introduced to either flow (figure 2-d)”.

4.5 Other Data Movement Patterns

- *Dependent target:* Often, data needed in a target location (T), partly exists in several sources. If the business processes of the T depend on the data of the sources, then the cost of transformation is usually done on the T side. When data moves from multiple sources ‘S1’, ‘S2’, etc. to a target ‘T’, and the T depends on the data in S then a cost of extraction, transformation and integration appears in each of the flows, possibly at the T side (figure 2-e).
- *Missing flow:* Often, there is a technical or governance barrier introducing a prohibitive cost that obstructs the implementation of the flow. Data needed by a consumer exists at a S, but are not able to reach the consumer (figure 2-f).
- *Ephemeral flow:* is a flow from S to T that exists for a short period of time (i.e. migration purposes) and is planned to be deleted in the future. Ephemeral flows are often created cheaply, with a short-term mindset, but then become part of the system, leading to future costs and complexity (figure 2-g).
- *Data movement:* Whenever data moves from its source to a destination there is the accumulative cost of extracting, transforming and loading the data from the source to the target. The cost might include staff training and support, and can be in either side of the flow (figure 2-h).

As shown in the patterns, costs and risks are likely to arise when data is moved between two entities that differ in some key way. When data is moved from producer (or holder) to consumer, it typically needs to be transformed from one format to another. Data values that make sense in the producer environment need to be converted into values that will be interpreted equivalently in the consumer environment. However, this conversion process is often difficult to apply correctly and completely, as the knowledge that is required is often stored tacitly in the heads of the data producers and consumers, rather than being explicitly declared in an easily accessible form. Where data is sensitive (as health care data often is) there are also governance issues to be considered. Data often cannot be shared unless it has been appropriately aggregated or otherwise anonymised.

Data may need to be filtered before it is moved, or moved through a particular set of systems, purely to be cleared for export to the real data consumer.

The outcome of our analysis of the case studies is a hypothesis regarding the features of a proposed IT development, and the socio-technical environment in which it is to be implemented, that act as early-stage indicators of implementation risk. We have designed a method for modelling just the parts of an organisation’s IT infrastructure that is needed to detect these features. In the next section, we present this model, and the method we have designed for constructing and using it to predict the risk of a proposed new IT development.

5 The Data Journey Model

Having characterised the kinds of data movement that can be problematic, the next step is to create a method for identifying the presence of the movement anti-patterns in a new development. In this section, we describe the modelling approach we have designed, aimed at capturing only the information needed to discover the movement patterns.

The core requirement is to identify the points in an information infrastructure where data is moved between two organisational entities which differ in some way significant to the interpretation of the data. These are the places where the portability of the data is put under stress, where errors can occur when the differences are not recognised, and where effort must be put in to resolve the differences. The model must therefore allow us to capture:

- The movement of data across an information infrastructure, including the entities which “hold” data within the system, and the routes by which data moves between them; we call this the model’s landscape.
- The points at which key differences in the interpretation of data occur, both social and technological.

5.1 The Model

We model the information infrastructure of an IT development in terms of the existing data containers, actors and links of the data journey model landscape. We use *data containers* to note the places where data rests when is not moving. A data container can be a system’s database whenever the data are in electronic form, or even a file cabinet, a pigeon hole, or a desk whenever the data are in a physical form. For example, when a general practitioner (GP) requests blood test results from the lab pathology of a hospital, data needs to travel from the GP secretary’s desk (where the request card and the blood sample rests), to the hospital porter’s pigeon holes, to the lab’s database (where results are input by the lab analyst), and back to the GP’s database to discuss with the patient. We model data containers using a rectangular box, as shown in figure 3.

Actors are the people or systems that interact with the containers to create, consume, or transform the data resting in them. In the example described above,

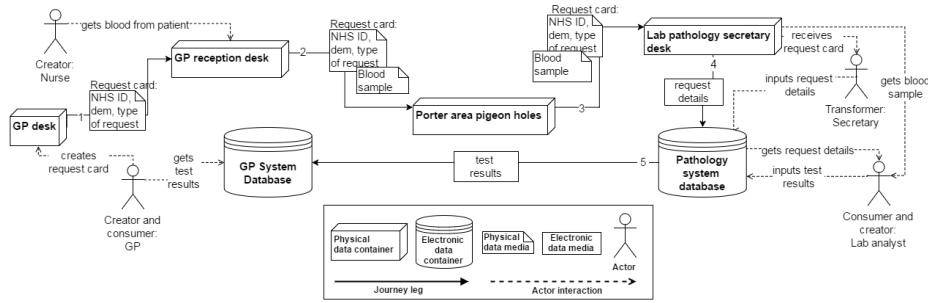


Fig. 3. Data journey diagram of a GP requesting blood test results from a pathology lab.

a lab analyst interacts with the lab system database to input the results of the analysis in. He is the creator of the test results data. The GP consumes those data by interacting with the GP system database. Actors are modelled using the actor symbol of the UML notation, and the interaction with the containers with a dotted arrow, as shown in figure 3.

While data may be stored in one container, it may be consumed at several places in the landscape. *Links* are the routes that currently exist between two containers along which data can move, and are modelled as straight lines between two containers.

To move along a link, data must be represented in a *medium* of physical or electronic form. For example, the request card resting in the secretary’s office is moved to the pathology lab by post. The test results move from the lab’s database to the GP’s system through an internet connection.

Containers, actors and links are parts of the landscape of the existing infrastructure in which data moves. Often, a new movement must be implemented. A *journey* describes the movement of data that needs to occur for a piece of data that is needed by some consumer to move from its point of entry into the landscape, to its point of use by the new actor. A data journey begins from a container storing the source data, and ends at the container which the end consumer interacts with. In figure 3, the initial container of the journey is the GP desk and the final consumer is the GP.

Sometimes a direct link between the source and target container doesn’t exist making the data to move through intermediary containers using existing links. Those intermediate links are called *legs*. A data journey is made up of a number of journey legs. Journey legs are modelled with an arrow connecting the containers in which data are moved between. The direction of the arrow shows the direction in which data needs to move. Journey legs can constitute existing links or create a new link between two containers.

Figure 4 shows the meta-model for the data journey model, expressed in UML. A data journey diagram is a set of consecutive journey legs. A journey leg moves a piece of data from a source container to a target container through

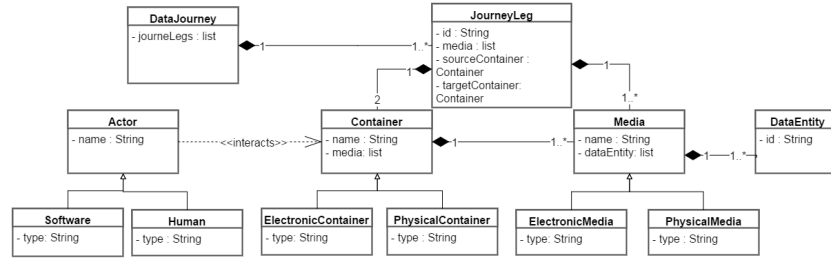


Fig. 4. The meta model of the data journey.

an electronic or physical medium. An actor interacts with a container to create, consume or transform the data stored in it.

5.2 Identifying Potential Costs

Having created a data journey model, the next step is to add in the information that can help us identify the legs where high cost or risk might be involved. We have seen from the case study analysis that costs and risks arise when data is moved between two entities that differ in some key way. Thus, when a human enters data into a software system, or two humans with very different professional backgrounds share data, or when software systems designed for different user sets communicate with each other, there is the potential need to transform or filter the data, to make it fit for its new context of use. However, to predict those places where costs might appear, we need cheap to apply information, since there is little value in predictions that cost a significant fraction of the actual development costs to create. We therefore focus on obtaining only the bare minimum of information needed, and ideally only on information that is readily available or cheap to acquire.

In the case studies, we found that high cost and risk occurred when data was shared between actors and containers with the following discrepancies:

1. Change of media: Containers using different media. For example, when a legacy X-Ray image on film must be scanned into a PDF for online storage and manipulation.
2. Discontinuity - external organisation: Containers belonging to different organisational units. For example, cancer data captured by a F.T needed for researching purposes by another agency.
3. Change of context, clash of grammars [9]: People speaking different vocabularies. For example, when a secretary is asked to transcribe notes dictated by a consultant.

We need low cost ways of incorporating these factors into the data journey model. In some cases, the information is readily available. For example, it is normally well known to stakeholders when information is stored on paper, in a filing cabinet, or in electronic form. However, other factors, like people's vocabularies,

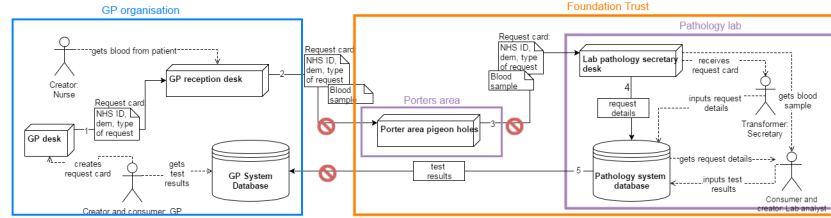


Fig. 5. Organisational boundaries and costs of the data journey diagram.

are less obvious. For these factors we use a proxy; some piece of information which is cheap to apply, and approximates the same relationship between the actors and containers as by the original factor. For example, we use salary bands as a proxy indicator for the presence of “clash of grammars”, on the grounds that a large difference in salary bands between actors probably indicates a different degree of technical expertise.

We use the following rules and proxies for indicating the presence of a boundary between the source and target of a data journey leg. A boundary indicative of high cost/risk can be predicted to be present when:

- the medium of the source container of a journey leg is different from the medium of the target,
- the source container of a journey leg belongs to a different organisational unit from the target container, or
- the actor creating the data at the source container has a different salary band than the actor consuming it at the target.

To identify the places in which the above factors may impose costs, we group together the elements of the data journey diagram with similar properties. For example, we group together all physical containers, or electronic containers, or clerical staff, clinical staff, elements belonging to the radiology department of a Foundation Trust (F.T.), elements belonging to the GP, and so on. These groupings are overlaid onto the landscape of the data journey model and form boundaries. For example, figure 5 shows the containers belonging to the GP organisation with blue colour and the ones belonging to the F.T. with orange colour. The places where a journey leg crosses from one grouping into another are the predicted location of the cost/risk introduced by the external organisational factor. In figure 5, the costly journey legs are noted with a red warning sign.

Other boundaries stemming from factors other than those stated above, are also likely to exist. However, we do not include them in this analysis since the amount of work needed to evaluate is another paper of its own. Both the boundaries described above and the data journey model have been evaluated in a retrospective study of a real world case study from the NHS domain. The study describes data moved from a GP organisation to the radiology department of a F.T.. The results of the evaluation showed that our model can identify places of high costs and risks. A further description of the results is given in [4].

6 Conclusions

In this paper, we have presented and motivated a new form of enterprise modelling focused on data journeys, which aims to provide a lightweight and reliable means of identifying the social and technical costs/risks of a planned IT development. Our approach is based on lessons learnt from case studies written by experienced NHS staff. The case studies showed how complex data movement can be in large organisations, and the numerous barriers that exist that introduce unexpected costs into seemingly straightforward data movement.

We have evaluated the effectiveness of the data journey approach, through a retrospective study of data movement in a nearby hospital trust. In this study, new software was brought in to reduce the costs of an existing data movement. Our approach was able to predict all the changes made by the development team, as well as proposing further improvements that the domain expert agreed looked promising. The details of the evaluation can be found elsewhere [4].

However, further evaluation of the approach is needed to fully test the hypothesis, and especially to test it in contexts that go beyond the healthcare setting of the case studies from which it was developed. In addition, we wish to explore further modes of use for the technique, since it is potentially capable of highlighting cost saving opportunities in existing systems, and of assessing organisational readiness to comply with new regulations (such as clinical care pathways and guidelines).

A The Failure Factors

People-Oriented Factors
<i>Resistance to change:</i> new IT systems can require staff to follow new processes and procedures, which they may resist. Staff can also fear replacement.
<i>Reluctance to engage with new technology:</i> staff may resist a change from familiar to unfamiliar technologies.
<i>Varying IT literacy levels among staff:</i> this can reduce or delay the benefits of the new IT systems, until the necessary skills are acquired.
<i>Clash of grammars[9]:</i> Specialist vocabulary used by, e.g., clinicians may be misunderstood by, e.g., secretaries, leading to delays and data quality problems.
<i>Lack of trust:</i> staff often don't trust the IT implementation team and don't share knowledge, domain expertise and ideas with them.
<i>Incorrectly identified stakeholders:</i> key users might not be identified as stakeholders and then will not be involved in decision making.
<i>Insufficient stakeholder engagement:</i> communication and engagement may vary between stakeholder groups (e.g. secretaries, clinicians, GPs, developers).
<i>Lack of a shared vision between stakeholders</i>
<i>Insufficient resources for support and training:</i> lack of training and support can cause resistance to use the new system, thus reducing/delaying benefits.

Data-Oriented Factors
<p><i>Numerous data sharing agreements:</i> records may be subject to many sharing agreements, each controlling a different, narrowly defined subset of clinical data.</p> <p><i>Conflicting formats/data structures :</i> data needs to be transformed before it can be used, often requiring substantial manual input.</p> <p><i>Disconnected data silos:</i> clinicians may trust their own legacy databases more than shared ones, leading to duplicated, fragmented and inconsistent data.</p> <p><i>Paper-based data:</i> Paper-based data transfer is still widespread, and brings many additional costs to a project.</p>
System-Oriented Factors
<p><i>System extensibility not considered during development:</i> extending an existing system with extra functionality can require significant reconfiguration.</p> <p><i>Over-optimistic system reuse:</i> when a system designed for one user group is adopted by another, major reconfiguration/process redesign may be needed.</p> <p><i>Inadequate system performance:</i> system performance problems, especially over the long term, can have a de-motivating effect on users.</p> <p><i>User-unfriendly system design:</i> new IT systems can be complex and inflexible. Systems may be wholly or partially unused, or workarounds may be needed.</p> <p><i>Costly system sustainability and maintainability:</i> systems that cannot change over the long term become decreasingly useful.</p> <p><i>Top-down system design:</i> managers may take decisions without consulting the end-users. The needs of specific regions, departments or users may not be met.</p> <p><i>Lack of required equipment:</i> if insufficient or inadequate equipment is provided, stakeholders may not receive the expected value from a new development.</p>
Organisation-Oriented Factors
<p><i>Organisation has a complex structure:</i> large organisations are typically coalitions of sub-groups, each with its own politics, culture, technologies and structures.</p> <p><i>Regulations:</i> organisational policies may conflict with the new system, and vice versa, leading to reduced/delayed benefits and need for reconfiguration.</p> <p><i>Organisational redesign:</i> IT developments often demand organisational restructuring, which is time consuming and often unwanted by staff.</p> <p><i>Resistance to collaborative projects:</i> staff may be reluctant to participate in IT projects, fearing loss of data/process ownership, control and influence.</p>
Governance-Oriented Factors
<p><i>Governance:</i> organisations may resist sharing data due to concerns over information and corporate governance, data ownership, privacy and confidentiality.</p> <p><i>Responsibility:</i> it may be unclear which organisational unit is responsible for developing/procuring and maintaining new systems, and who owns the data.</p>
Requirements-Oriented Factors
<p><i>Design-reality gap:</i> the new system may not work as envisaged, or may contradict processes and practises previously in use. Workarounds must be devised.</p> <p><i>Engagement:</i> if the implementation team doesn't engage with clinicians and other key stakeholders, the system functionality may not meet user needs.</p> <p><i>Effect on quality of care:</i> the need to input new data cuts in time that would otherwise be spent focusing on the patient, and can reduce the quality of care.</p>

Politically-Oriented Factors
<i>Loss of political influence:</i> a department may fear loss of influence and power when a new IT system is introduced.
<i>Financial pressures:</i> projects are often expected to deliver with a tight budget/insufficient resources, leading to expensive and risky workarounds.
<i>Varying political power:</i> the functionality that is built may be determined by those with the most influence, rather than the actual end users.

References

1. R. S. Aguilar-Saven. Business process modelling: Review and framework. *International Journal of production economics*, 90(2):129–149, 2004.
2. B. Boehm, C. Abts, and S. Chulani. Software development cost estimation approaches - a survey. *Annals of software engineering*, 10(1-4):177–205, 2000.
3. Y. L. Chen et al. Data flow diagram. In *Modeling and Analysis of Enterprise and Information Systems*, pages 85–97. Springer, 2009.
4. I. Eleftheriou, S. Embury, R. Moden, P. Dobinson, and A. Brass. Data journeys: Identifying social and technical barriers to data movement in large, complex organisations. Technical report, School of Computer Science, The University of Manchester, 2016. Access link: www.datajourney.org/publications/tech_rep_data_journey.pdf.
5. J. P. Marshall. The social (dis)organisation of software: Failure and disorder in information society. *The Australian Journal of Anthropology*, 25(2):190–206, 2014.
6. T. Pardo, A. M. Cresswell, S. S. Dawes, G. B. Burke, et al. Modeling the social & technical processes of interorganizational information integration. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, pages 8–pp. IEEE, 2004.
7. J. Rumbaugh, I. Jacobson, and G. Booch. *Unified Modeling Language Reference Manual, The*. Pearson Higher Education, 2004.
8. Y. L. Simmhan, B. Plale, and D. Gannon. A survey of data provenance in e-science. *ACM Sigmod Record*, 34(3):31–36, 2005.
9. J. L. Vann. Resistance to change and the language of public organizations: A look at “clashing grammars” in large-scale information technology projects. *Public Organization Review*, 4(1):47–73, 2004.
10. E. S. Yu. Social modeling and i*. In *Conceptual Modeling: Foundations and Applications*, pages 99–121. Springer, 2009.
11. M. M. Yusof, J. Kuljis, A. Papazafeiropoulou, and L. K. Stergioulas. An evaluation framework for health information systems: human, organization and technology-fit factors (hot-fit). *International journal of medical informatics*, 77(6):386–398, 2008.
12. L. Zheng, T.-M. Yang, T. Pardo, and Y. Jiang. Understanding the “boundary” in information sharing and integration. In *Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS’09)*, pages 1–10. IEEE, 2009.