

# Discovering taxonomic structure in design archives with application to risk-mitigating actions in a large engineering organisation

Chuck Hsiao<sup>a</sup> , Michael Ruffino<sup>a</sup>, Richard Malak<sup>a</sup>, Irem Y. Tumer<sup>b</sup> and Toni Doolen<sup>b</sup>

<sup>a</sup>Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA; <sup>b</sup>School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, OR, USA

## ABSTRACT

This paper demonstrates a general iterative technique for discovering a taxonomy that categorises information contained within a weakly structured design project archive. It is common for large engineering organisations to maintain archives about past design projects. In principle, engineering organisations could mine these archives to extract useful lessons and relationships that would improve future design projects. However, this is difficult in practice, since archives often consist of weakly structured data formats, such as plaintext documentation. This restricts the use of many useful analysis tools. The taxonomy-discovery process presented here is a critical first step towards unlocking the value of such archives. The technique is based on methods from the qualitative research literature. We demonstrate this process by creating a taxonomy of risk-mitigating actions in design projects based on a design project archive from a large engineering organisation. We discuss practical considerations such as missing contextual information as part of the case study. The taxonomy is sufficiently generic to be of use to other organisations. Furthermore, individual organisations can use the iterative technique introduced in this paper to tailor the taxonomy to their own project archives. Thus, this research provides an important foundation for unlocking the value of archived design project information.

## ARTICLE HISTORY

Received 21 February 2014  
Accepted 29 November 2015

## KEYWORDS

Design information management; knowledge representation; taxonomies; risk analysis and management

## 1. Introduction

Engineering organisations, particularly larger ones, tend to record a great deal of information about their design projects. In addition to documenting the design specification itself, these organisations often record information about events that occur during a design process, decisions they make and the rationale for them, and the outcomes of specific courses of action (Heisig, Caldwell, and Clarkson 2014). Motivations for recording this information can include the desire to support the management of active projects, to bring new personnel up to speed on a project, and to maintain compliance with government regulations.

Information about past design projects can be valuable to engineering organisations in many ways. It provides an opportunity to compare actual practice to intended practice; it

can be used to identify best practices, effective methods, and new engineering tools; and it can highlight the need for improvements even if it does not solve the problem itself. Information about the performance of the design artefact from previous design projects can also aid in the design of better artefacts in current projects (Jagtap and Johnson 2011), and reduce the cost of creating new parts (Yu, Cha, and Lu 2012). Although engineering organisations recognise the potential value of archived design project information, there are many practical challenges to exploiting this resource (Salzberg and Watkins 1990; Hertzum and Pejtersen 2000; McMahon et al. 2004). Foremost among these is that design project archives typically are limited in structure and semantic content. Many consist of little more than plaintext documentation or loosely structured information, such as text entered into fields in a form (Cheung, Lee, and Wang 2005). This impedes the application of design informatics tools that could discover useful insights and relationships in the data. Nevertheless, some researchers have found success in using such tools in unstructured organisational information, such as categorising concepts and intellectual capital found in organisational emails (Cheung et al. 2011; Tsui et al. 2014). Others have researched using text mining techniques to extract design rationale from free text documents (Liang et al. 2012). Simply categorising the information in weakly structured archives can help an organisation better identify and understand past and current actual practices, which can aid in formalising and systemising engineering procedures, improve engineering training, and facilitate more efficient record-keeping. For example, categorising past ad hoc actions taken to solve a category of engineering problems may provide for the creation of a checklist of actions for that type of problem, helping engineers more quickly determine the actions needed for a solution, and determining if it has been carried out (Catic and Malmqvist 2013).

In this paper, we demonstrate an iterative technique for discovering taxonomic structure in minimally structured design project information. An engineering organisation seeking to better understand its engineering practices using archival information, or to evaluate the likelihood of finding useful knowledge in such archives, can use this technique to better characterise and organise that information.

The proposed taxonomy-discovery technique is rooted in established methods from the qualitative research literature. The technique requires an initial taxonomy to seed the process. This can be chosen from related literature or based on a preliminary analysis of the archive. Two or more individuals – termed *coders* – modify the taxonomy iteratively until they reach a point of convergence, which is defined by achieving a satisfactory level of inter-rater reliability as measured by Cohen's kappa. A validation step using an additional individual – termed the *naïve coder* – ensures that the success of the primary coders is due to the taxonomy rather than their own tacit understanding of the archive.

We demonstrate the general taxonomy-discovery technique through a case study based on an archive from a large engineering organisation. The purpose of the archive is to record risks encountered in design projects and actions taken by engineers to mitigate those risks. The archive consists of 185 risk reports containing 822 action/event entries. Using our method on this archive, primary coders reached a consistency level of 62.2% and a Cohen's kappa of 0.556 after seven taxonomy-refinement iterations. The naïve coder achieved a match rate of 68.3% and a kappa of 0.628 relative to the primary coders. These agreement levels compare well with results typical in the qualitative

research field and suggest that the proposed method is an effective way to identify the structure of information locked in a pre-existing engineering project archive.

A secondary contribution of this work is the taxonomy generated during the case study. It was developed based on an archive from an engineering organisation, and so it is of the most use to that particular organisation. However, it provides additional insight into risk-mitigating actions in engineering projects and can serve as the seed taxonomy for applying our proposed method to similar archival information within other organisations. The taxonomy allows for the classification of actions intended to mitigate *project risk*, which is an important aspect of any large design project. The engineering literature contains many studies related to *technical risk* (i.e. likelihood and consequences of failure in the system being developed) but relatively little study of risk at the project level. The final taxonomy classifies risk-mitigating actions using two attributes – purpose and embodiment – each decomposed into several categories.

The remainder of the paper is organised as follows. Section 2 is a brief review of taxonomy generation from information archives and the use of Cohen's kappa as a validation metric. Section 3 describes the taxonomy-discovery process. Section 4 details the demonstration of the process, including a background on risk, an overview of the archive used, the main steps taken to refine a seed taxonomy based on the iterative coding process, and a description of the final taxonomy of risk-mitigating actions. Section 5 describes the taxonomy validation results. Section 6 discusses the broader implications of this work and further topics for research. Section 7 summarises the conclusions developed as a result of this research.

## 2. Taxonomy generation

### 2.1. Foundations and general process

A taxonomy is a hierarchical arrangement of concepts (Gilchrist 2003). It can be useful for classification purposes and can be combined with relationships to form an *ontology*. Taxonomy generation is typically an expert-driven process in which taxonomy authors are the experts themselves (Carr et al. 1993; Kurtoglu et al. 2009) or experts are consulted via an interview process (Ahmed, Kim, and Wallace 2007). However, this may be impractical in many cases. First, an appropriate subject matter expert may not be available for various reasons (e.g. personnel knowledgeable about design practices within the company at the time may no longer be with the company). Second, actual archiving practice can deviate from the ideals that would be identified by experts. Design archives can contain textual or weakly structured information contributed over long periods of time by numerous individuals who are given little direction regarding archival standards. It is difficult to enforce consistency in semantics and quality in such a situation (Vermaas 2013), and it often is impossible to contact authors of specific entries later for clarification (e.g. because they no longer work for the organisation). Thus, expert-driven or interview-driven taxonomy generation processes may not guarantee success in these types of archives. In contrast, the method proposed in this paper extracts taxonomic structure directly from the archive and therefore is robust to this effect. In the event that experts are available to generate a taxonomy, the proposed method also can be used to independently corroborate or further refine the expert's taxonomy.

An alternative to expert-driven taxonomy generation is to discover a taxonomy using the information archive itself. Recent research has focused on the development of rules for automated generation of a taxonomy from a set of texts or corpus (Paukkeri et al. 2011; De Knijff, Frasinicar, and Hogenboom 2013; Medelyan et al. 2013; Ceesay and Hou 2015), and especially to build *ontologies* which can include additional relational concepts between nodes, compared with the simple hierarchical ‘parent/child’ relationship of a taxonomy, such as in Li et al. (2012). However, the costs involved – not only in terms of computational resources but also in terms of manual effort in generating and testing different rules – may be substantial (Simperl et al. 2012). Building ontologies from a relatively weakly structured data source can also be problematic (Hepp 2008). Using ontologies for information retrieval typically involves formal query languages, requiring the user to have knowledge of the domain ontology as well as the syntax of the query language (Kara et al. 2012). Additionally, a taxonomy of a set of data can be used as a precursor for building an ontology of that data, in the event that an ontology is judged likely to be useful, such as in Li et al. (2014). Thus, in this paper we focus on the manual development of a taxonomy from an information archive.

In order to manually generate a taxonomy from the information archive itself, we adapted a technique based on the concept of *coding* from the qualitative research community (Auerbach and Silverstein 2003). In qualitative research, coding is the process of segmenting a set of data, typically textual data, and then annotating each segment with computer-interpretable codes. Coders use a taxonomy of possible codes and their definitions to assign codes to each segment. Once completed, codes for each segment can then be analysed statistically. Coding can be applied to interviews, reflexive journals, documents, and a variety of other media where information primarily is in textual form.

We use coding in this research in two ways. First, it serves as part of an iterative process of taxonomy refinement. This is described further in Section 3. Second, coding serves a validation role through the use of a *naïve coder*, so called because this individual is not involved in the taxonomy-refinement process (Ahmed, Kim, and Wallace 2007). This step is important because it verifies that the taxonomy represents the coders’ understanding of the categories, independent of their level of match agreement. For example, they may have tacitly classified a given situation as a particular category and thus achieve high agreement levels, even though the situation is not specified in that category’s description. Thus, if a naïve coder exhibits low levels of agreement with the primary coders, it is an indication that the taxonomy is lacking key concepts or that their meanings are not defined clearly. Conversely, high levels of agreement indicate that the taxonomy is an adequate explicit formalisation of the categorical relationships among concepts in the information archive.

## 2.2. Measuring coder agreement

Agreement among two coders in this research is measured using Cohen’s kappa (Cohen 1960; Berry and Mielke 1988; Brennan and Hays 1992; Cantor 1996). It is a widely used and widely studied metric in qualitative research for inter-rater agreement. This measure is related to the match rate, the number of matches between two coders divided by the

total number of entries. However, it takes into account the match rate that would occur due to random chance. Mathematically, Cohen's kappa is defined as

$$\kappa = \frac{\Pr(a) - \Pr(r)}{1 - \Pr(r)}, \quad (1)$$

where  $\Pr(a)$  is the actual observed match rate between the two coders, and  $\Pr(r)$  is the match rate due to random chance. The calculation of  $\Pr(r)$  takes into account the distribution of codes assigned by each coder. That is, it does not assume coders assign each code with equal probability, but accounts for different amounts of each code assigned by each coder.

Kappa is unity if both coders match on every entry, and zero if their match rate is no better than random. Kappa will be negative if the match rate is lower than what would be expected based on chance.

Although there is some disagreement over what values of kappa indicate good inter-rater reliability (Bakeman et al. 1997), generally values around 0.5 are considered fair and values above 0.7 are considered good (Landis and Koch 1977; Robson 1993). In this sense it is similar to Pearson's correlation coefficient  $r$ , in that although the metric has a clear mathematical definition, how high of a value to expect for data to be considered reliable is open to interpretation.

Published values of kappa vary. For example, Adler and Johnson (2000) reported a kappa of 0.49 when applying a taxonomy to papers in the field of computer-aided instruction in medical education. Spooren, Mortelmans, and Denekens (2007) reported kappa values ranging from 0.228 to 0.567 (and a Pearson's correlation coefficient  $r$  ranging from 0.487 to 0.861) for students taking a teacher evaluation questionnaire twice, as a measure of each student's intra-rater reliability. Wilcox-Herzog (2004) reported a kappa of 0.75 and 0.79, respectively, in a study evaluating the verbal responsivity and teacher involvement of early childhood teachers.

The advantage of using Cohen's kappa over the common Pearson's correlation coefficient  $r$  is that Cohen's kappa can be used on categorical data, such as in this study, whereas Pearson's coefficient  $r$  can only be used on ordered or numerically valued data. Furthermore, Cohen's kappa measures the match rate between coders, while Pearson's coefficient  $r$  measures the correlation. In some circumstances, two observers can have a high correlation even when their match rate is low. An example is if one grader consistently gave 10 points more than another grader when scoring tests. In that case Pearson's coefficient  $r$  would report a high correlation, which though true is misleading, whereas Cohen's kappa would give a low value for the inter-rater reliability (Clark-Carter 1997), highlighting the discrepancy between the graders.

### 2.3. Example: taxonomies related to risk

The taxonomy-refinement technique presented in this paper will be demonstrated by classifying entries in an archive of risk-mitigating actions from a large engineering organisation. Although taxonomies of risk and risk management exist, they focus on risk management in general and not specifically on risk mitigation (Boehm 1991; Muehlen and Ho 2006; Aloini, Dulmin, and Mininno 2007). Thus, although such taxonomies are useful for determining the best risk management approach to use in a given situation, they are less useful for

identifying specific actions to pursue once risk mitigation is chosen. For example, Miller and Lessard (2001) discuss under what conditions risk mitigation should be applied versus other types of risk treatment, such as diversifying the types of projects that an organisation pursues, but do not go into detail beyond a brief description of risk mitigation. Peltier (2004) (using different terminology) gives a brief description of the types of risk treatment options, but also does not go into detail. By contrast, this paper focuses specifically on the types of actions taken when applying a risk mitigation treatment.

### 3. Iterative taxonomy-refinement method

The taxonomy-refinement method starts with a seed taxonomy to initiate the refinement process. Coders can identify a seed taxonomy from related literature or can devise one based on the available domain expertise. If a seed taxonomy is not available, the coders can use several archive entries as the basis for one. Intuitively, a better seed taxonomy should lead to a refined taxonomy more quickly. Thus, it is worth investing some effort in identifying a good seed taxonomy.

The archive is segmented into individual entries. The coders then code multiple entries independently using the current taxonomy. They compare the codes that they gave for each entry. The degree to which they agreed on how to classify entries in the section serves as a measure of taxonomy effectiveness.

For codes that do not match, the coders discuss potential reasons for the mismatch. Because the categorisation process consists of coders using a taxonomy to categorise archive entries, the three possible sources of error are the taxonomy, the coders themselves, and the archive. If the taxonomy is the reason, they modify the taxonomy such that only one set of codes would be appropriate for that segment of text, so that similar segments do not cause mismatches. For example, the coders may clarify and revise the definitions of the categories in the taxonomy if the definitions are too narrow or overlap each other. If the reason for the mismatch is due to coder misunderstanding, then the coders discuss if the taxonomy should be clarified to prevent such misunderstandings. If the reason for the mismatch is due to the archive entry itself (e.g. it lacks sufficient contextual information), then the coders may not need to change the taxonomy. The process then repeats on a new subset of the archive using the revised taxonomy.

During taxonomy refinement, some archive entries may not clearly fit any of the existing category definitions. In this case, coders may choose to place them in an 'other' category. The coders should periodically discuss the entries in this category to see if existing categories should be redefined to contain them, or if a new category is needed for them. Also, the coders should discuss whether any categories contain an unusually large proportion of entries. This may be an indication that the category should be split into multiple categories with more specific definitions. This ensures that the taxonomy retains its descriptive power on the data.

Once the coders are able to independently code many entries using the taxonomy with high agreement, they then introduce the taxonomy to a naïve coder. Other than the basic process of coding, they should not give too detailed instructions to the naïve coder, and particularly not about the definitions of the terms in the taxonomy. This ensures that the naïve coder uses the definitions as given by the taxonomy, rather than as communicated directly by the original coders. The naïve coder then codes multiple entries which the

original coders had already coded, and the results are compared. If the naïve coder's results exhibit high agreement with the coders', then this indicates that the taxonomy can adequately describe the archive entries and that the taxonomy-refinement process is complete. If not, then the taxonomy definitions should be clarified and this process repeated, as it indicates that the high agreement between the original coders may be due to their tacit understanding of the categories rather than what is explicitly captured by the taxonomy's definitions.

Figure 1 illustrates the taxonomy-discovery process developed in this research.

## 4. Demonstration: discovering a taxonomy of risk-mitigating actions

### 4.1. Engineering project risk

The demonstration of the taxonomy-refinement method is on an archive of risk-mitigating actions from a large design organisation. Managing project risk is an important undertaking in any large, complex design endeavour. Risk in this context is a state of uncertainty where some possible outcome can have an undesirable effect (Thunnissen 2003). Thus risk has two

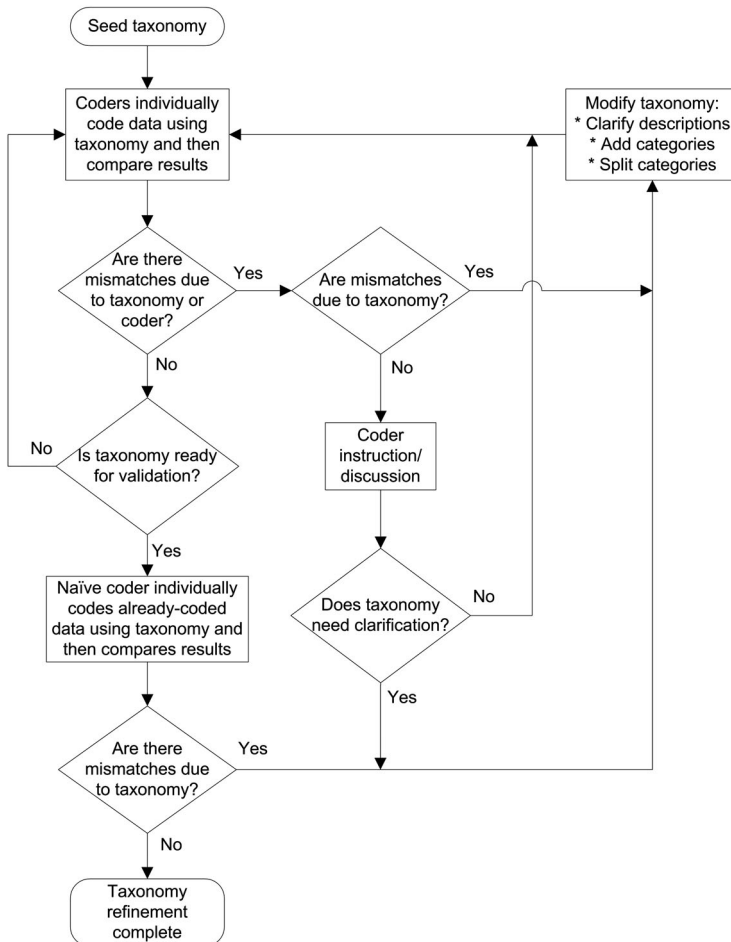


Figure 1. Flow chart of taxonomy-discovery technique.



major components: (1) uncertainty, where limited knowledge results in an inability to accurately and precisely describe the future outcome or state, and (2) undesirable effects, where an outcome may affect the project negatively. Inadequate treatment of risks at the project level can result in wasted money, effort, and time, in addition to secondary negative effects such as damage to an organisation's reputation and diverting manpower and resources from other projects. Consequently, there exists strong motivation to provide decision-makers in engineering projects with models and information that they can use to predict how different actions will impact relevant project metrics.

In much of the prior literature on engineering risk, the focus is on technical risk in terms of the likelihoods and consequences of potential system failures. In contrast, project risk relates to higher level considerations such as project cost and schedule. Although technical risk can be a strong indicator of project risk, the two are not synonymous. For example, an organisation could use parallel development teams on a subsystem objective to reduce project risk, even though the technical risk is not reduced.

Techniques for modelling technical risk in engineered systems tend to be based on bottom-up approaches that require detailed information about individual components and how they interact within a system. Examples include Failure Modes and Effects Analysis; Failure Modes, Effects, and Criticality Analysis (Stamatis 2003); and probabilistic risk assessment techniques such as Fault Tree Analysis and Event Tree Analysis (Vesely et al. 1981; Greenfield 2000; Bedford and Cooke 2001; Stamatelatos and Apostolakis 2002). Given the complexity of many large-scale design projects, it may be impractical to develop models of project risk using a bottom-up approach analogous to those used for technical risk.

An alternative approach is to model project risk using data from a representative set of past design projects. This is viable if there exists a suitable source of data upon which organisations can base their models. Although large engineering organisations record much information about their design projects, they do not always do so with an eye towards data-driven modelling. As such, engineering project archives and documentation can consist largely of unstructured text with inconsistent standards for terminology and informational detail (Salzberg and Watkins 1990; Lowe et al. 2000). At present, it is unclear to what extent unstructured legacy project data are useful for developing models about project risk or other factors of interest to engineering decision-makers. This research is an important step towards answering this question.

Once a risk has been identified and analysed, the treatment of the risk can generally be categorised into four different types of actions: risk avoidance, risk sharing, risk mitigation, and risk acceptance (Muehlen and Ho 2006). In a design project where an organisation has direct control over the risks, risk mitigation is often the most appropriate risk treatment strategy (Miller and Lessard 2001). Identifying what types of risk mitigation have the most impact on different projects would help decision-makers plan for a project, and categorising risk-mitigating actions into a taxonomy is an effective step towards determining which actions are the most useful.

#### 4.2. Data source

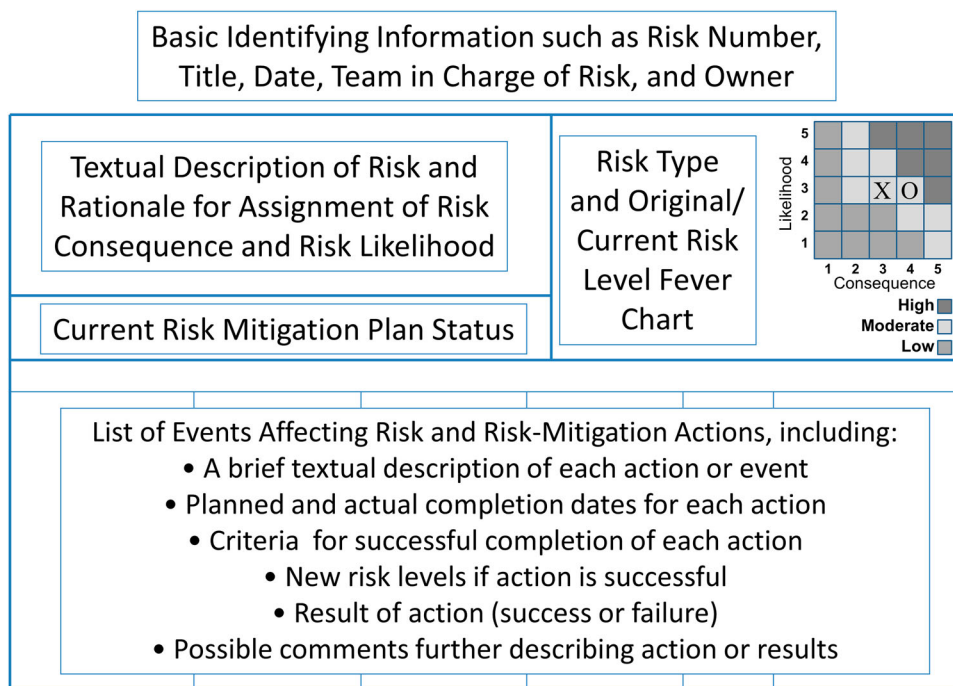
The taxonomy in this paper was created using a legacy archive of actions taken during different design projects within a single organisation. This represents a different approach from using experts to generate the taxonomy. Because the archive entries were created



prior to this research study, the engineers who generated the data in the archive were not available to clarify the text.

Furthermore, the project archive used in this research largely recorded the actions taken during a project and not the reason(s) for each particular course of action. Thus, the archive is a design history archive as identified by Rockwell et al. (2010), rather than a design rationale archive. The lack of data related to the motivations and reasons for undertaking specific actions presented additional challenges for this research related to refining the taxonomy, since engineering archives that do not preserve design rationale can make the archive more difficult to understand and interpret (Regli, Kopena, and Grauer 2011).

The archive used for this study was created by members of a large engineering design organisation, which designs and manufactures highly complex systems and has over 50,000 employees. It contains reports of issues that were determined to have a substantial risk related to a subsystem of a particular product model, and the actions undertaken by engineers to mitigate that risk. Each risk report contains an initial numerical assessment of the risk's likelihood and consequence, and entries that detail the actions performed to mitigate the risk in textual format, as well as events that occurred affecting the risk. Estimates of new risk levels (likelihood and consequence) after each action or event, assuming that the action or event would succeed or come to fruition, were also given. Each risk report also contains a textual description of the risk and justification for the risk assessment, as well as details such as the people in charge of the risk report, scheduled and actual completion dates of each action, success or failure of each action, and the criteria used to gauge the success or failure of each action. Figure 2 illustrates the general layout of each risk report.



**Figure 2.** Layout of a risk report entry.

Although one can perform rudimentary analyses of the archive as-is, such as schedule delay rates, more sophisticated analyses require categorisation of the actions taken, which is not provided in the archive and must be generated. For example, such analyses could extract more useful information, such as what types of actions or what sequences of actions were the most beneficial in a project, based on its current state when the actions occurred. A major roadblock to analysing the records has been how to categorise those actions. Although the information was archived presumably for future reference, lack of training and expediency may result in entries of varying quality. In other words, information for a particular risk report could be missing important details or contextual clues that might be useful for retrospective analysis. Within the archive, the available contextual information differed from report to report, with some reports being very terse and others being very detailed. This variability further complicates the refining of the taxonomy.

The archive used for this study contained 185 risk reports. Of those risk reports, 68 reports merely detailed the risk without describing any risk-mitigating actions and were thus excluded from this study. The remaining 117 risk reports contained 822 action or event entries.

### 4.3. Seed taxonomy for this study

The primary coders determined that a taxonomy based on action research in design by Coughlan and Coughlan (2002) was suitable as a seed taxonomy for this study. Though representative entries from the archive could also have been used to generate a seed taxonomy, this allowed for a more direct connection to existing literature on risk mitigation. Although the taxonomy was reasonably good as a starting point, coding mismatches led to its revision into a taxonomy that was better suited to this particular archive.

The seed taxonomy consisted of six categories: *data\_gathering*, *data\_feedback*, *data\_analysis*, *action\_planning*, *implementation*, and *evaluation*. *Data\_gathering* referred to the collection of information pertinent to a project, such as gathering statistics, observations, or conducting interviews. *Data\_feedback* involved sharing the data among stakeholders so that it was available for analysis. *Data\_analysis* described using various analytical tools on collected data to better understand a problem. *Action\_planning* involved considering different courses of action and then deciding on the best course to pursue. *Implementation* referred to carrying out various tasks detailed in a plan. *Evaluation* indicated reviewing the outcomes of the plan to evaluate the implementation's effectiveness.

### 4.4. Taxonomy refinement

As coders used the seed taxonomy and analysed the results, inadequacies in the seed taxonomy became apparent and the taxonomy was revised to better describe the legacy archive. The refinement of this taxonomy involved multiple iterations, from merging categories to splitting categories to creating new categories, and the development of a new attribute in the taxonomy. Changes to the taxonomy are described next in chronological order.

*Addition of event type.* The original taxonomy was for actions that could be undertaken by a stakeholder or someone who was involved with the project. However, the archive also documented occurrences that were undertaken by outside entities rather than the

stakeholders that affected the outcome of the project. Examples included certification by a regulatory agency and work performed by subcontractors. Such occurrences were labelled as *events* to distinguish them from *actions* over which the design organisation had direct control. This is because a future aim of this study is to understand how different actions that a design organisation may take can affect the outcome of a project, with the goal of being able to recommend different actions or action types in different situations. Although an organisation's actions may affect the outcome of events (such as whether or not the design artefact is certified), it cannot directly carry out those events. Therefore, events, other than being classified as such, were not categorised further.

*Removal of data\_feedback.* The dissemination of data to other stakeholders was observed only rarely in the archive. In the cases where it occurred, such as 'release of part design', the purpose was to mark the completion of analysis, rather than to communicate information. Thus, this category was removed.

*Addition of embodiment attribute.* It became apparent to coders that the taxonomy was capturing the purpose of actions, but not how they were executed. Certain actions could have different implications depending on how they are executed. For example, holding a meeting with another department to develop a plan may have different effects on a project than developing a plan internally. Thus, the taxonomy was expanded to have two attributes for an action, *purpose* and *embodiment*. The purpose contained the then-current set of categories, *data\_gathering*, *data\_analysis*, *action\_planning*, *implementation*, and *evaluation*. The embodiment described how an action was executed. Thus, each action code could have two components, such as *coordinate-data\_gathering*. Initially, the embodiment attribute included *plan*, *meet*, and *coordinate* as categories. *Plan* was preparation to perform an action. *Meet* was to meet and exchange information with other groups working on the same action. *Coordinate* was working together with other groups on the same action. There was also the option of leaving the embodiment blank, if no particular embodiment was noted.

*Refinement of embodiment categories.* As the embodiment of different actions was classified, changes were made to the categories in this attribute. *Meet* was confused often with *coordinate*, so *meet* was integrated with *coordinate* to describe when multiple groups worked together to perform an action. *Inform* was added to signify a one-way transfer of information, such as a training workshop. In some ways this is similar to the original category of *data\_feedback*. However, informing of a purpose carries more information than simply *data\_feedback*. For example, *inform-data\_gathering* could signify a training workshop, while *inform-evaluation* would mean transferring test results.

*Split of action\_planning into approval and plan\_or\_schedule.* As actions were classified, significantly more actions were classified as *action\_planning* than any other category, so this category was split into two categories. *Approval* was a common action due to the size and complexity of the design organisation; other departments or sometimes regulatory agencies had to be notified and sign off on a proposed action. Separating *approval* into its own category represents a departure from the original classification scheme. The second refined category was *plan\_or\_schedule* and it was related to the allocation of manpower, such as organising teams to solve a problem or planning out a testing schedule in advance.

*Further refinement of embodiment categories.* The use of *inform* for certain actions was problematic, so this embodiment category was split into two categories: *request* and the

original *inform*. *Request* is a one-way communication, but requires a response, whereas *inform* is a one-way communication that does not require a response. *Request* differs from *coordinate* in that *coordinate* connotes mutual, continual two-way communication. Thus, *request-approval* denotes submitting an application to another department or regulatory agency to be approved, while *inform-approval* (although rare) would be notification of an approval. Also, *inform-gather\_data* would mean the dissemination of information, while *request-gather\_data* would be to request information pertinent to a project from other departments or other sources. *Plan* overlapped with the purpose category *plan\_or\_schedule*, so it was removed.

Split of *plan\_or\_schedule* and *technical\_evaluation* into *resource\_planning*, *technical\_planning*, and *technical\_evaluation*. A large proportion of taxonomy errors were actions that had both planning elements and technical elements, such as setting a goal, which directs the allocation of resources but involves technical analysis for setting a feasible target. Thus, the category of *technical\_planning* was created to encompass actions that had both resource and technical elements, while *resource\_planning* involves manpower and scheduling issues and *technical\_evaluation* is for purely technical analysis such as CAD modelling.

#### 4.5. Final taxonomy

The taxonomy classified each risk report entry into one of two major types, *actions* or *events*. *Actions* are executed by people under the purview of the risk report, whereas *events* are not. This classification thus first separates entries into actions that can be executed by the people involved with mitigating the risk, and events which affect the risk mitigation process but cannot be executed by the stakeholders directly, although they can influence the outcome of the events. For example, running a simulation is an action that the design organisation can choose to undertake in the course of mitigating a risk. However, acceptance or rejection by a regulatory agency is an event that the organisation cannot directly carry out, although they can impact its likelihood of success. Because this research was focused on classifying risk-mitigating actions, events, other than being identified as such, were not decomposed further.

Actions have two attributes, *purpose* and *embodiment*. Thus, each action is codified into an embodiment–purpose tuple. The purpose describes how the action helps mitigate the risk within the scope of the risk report, while the embodiment captures how the action is carried out. Table 1 contains examples of each action category. The categories within each attribute are defined next.

##### 4.5.1. Purpose categories

The purpose attribute encapsulates an action's primary objective with respect to mitigating the risk. It describes how the action aids in decreasing the risk likelihood or the risk consequence, if successful.

*Gather\_data*. This category is collecting information to be used in a subsequent action. This can include researching prior work as well as making measurements to aid in decision-making. Examples include looking at previous lessons learned or making initial measurements of a system.

**Table 1.** Taxonomy of risk-mitigating actions.

Attribute	Category	Description	Example
Purpose	Gather_Data	Gathering prior knowledge or observations for future action	'Tabulate expenses related to project'
	Resource_Planning	Allocation of manpower and scheduling	'Form Weight Focus Team'
	Technical_Evaluation	Technical analysis and decision-making	'Perform trade study for method to reduce load'
	Technical_Planning	Planning requiring technical analysis	'Set weight reduction goal based on materials/structures analysis'
	Approval	Obtaining consent for an action from another body	'Program approval of load reduction method'
	Implementation	Executing a previously decided plan or process	'Install new part'
	Validation_and_Testing	Verifying effect of previous action or implementation	'Review material results'
	Other	Does not fit above categories	'Coordinate customer intro'
Embodiment	Inform	Primarily one-way transfer of information	'Hold a risk understanding meeting'
	Coordinate	Multiple departments working together on action	'Develop plan with X Department'
	Request	Transfer of data with expectance of reply	'Submit plan for approval'
	Typical	No special embodiment of action	'Determine temperature at point X.'

*Resource\_planning.* Primarily organisational, this category is for the management of manpower and/or for scheduling a timeline for future actions. Deciding on a course of action is also included in this category. Examples include forming a team to investigate an issue or scheduling a test.

*Technical\_evaluation.* This purpose category is for actions of a purely technical nature. This includes the generation of alternatives, analysis of alternatives, and choosing among alternatives. Examples include determining the stresses at a given point in a mechanical system or using CAD to find the optimum shape and thickness of a supporting beam.

*Technical\_planning.* This category is used when both resource management and technical analysis are involved. For example, setting a production goal not only involves the commitment of manpower and resources, but also involves technical analysis to ensure that the goal is feasible. This is differentiated from *approval* below in that the department(s) involved sets the goal directly, rather than relying on another department.

*Approval.* This category involves obtaining consent from another body or organisation, such as from another department or from a regulatory agency. Note that this category is related to the act of requesting approval; whether or not another body grants the approval, if listed as its own entry in a risk report, is often an event. An example is submitting a plan or analysis to a certification board.

*Implementation.* This category is putting into effect a fully developed plan or process. Examples include installing a redesigned part or uploading new control software into a system.

*Validation\_and\_testing.* This purpose category is for actions to verify that the effect of a previous action or implementation is as predicted. This generally takes place after implementation and ensures that the actual performance matches expected or calculated results. An example is testing to verify that a new design works as modelled in CAD.

*Other.* This category was used primarily during taxonomy refinement to set aside entries that needed further discussion because they did not fit into any other category.

This category typically had very few entries, but served primarily to ensure that there remained an option if an action did not fit any existing categories. Occasionally, if enough similarities between actions with the purpose *other* are found, they may form a new category.

#### 4.5.2. Embodiment categories

Occasionally, there is something noteworthy about how an action was carried out. In general, the embodiment attribute by default was *typical*, denoting no particularly noteworthy embodiment, but some particular types of how actions were carried out were recorded for future study.

*Inform.* This embodiment category indicates that the action is primarily a one-way transfer of information, from one entity to another. For example, a training workshop would be classified as *inform-gather\_data*.

*Coordinate.* This category indicates that the action is of a collaborative nature, with multiple departments working together on the same action. An example would be meeting with other departments to plan a schedule for the integration of a subsystem.

*Request.* This category applies when information is shared, and a response is needed. This was used most frequently for *request-approval* to indicate submitting a plan or testing results to another department or regulatory agency, or *request-gather\_data* to denote asking other departments for information concerning a risk

## 5. Taxonomy validation results

Inter-rater match rate and inter-rater reliability using Cohen's kappa between two coders using the taxonomy were used as a measure of taxonomy competence and clarity. In this context, 'competence' refers to the ability of the taxonomy to encompass all coded actions, and 'clarity' refers to the distinctness of the definitions of categories relative to each other. To test if the taxonomy fully captures the information used by the coders, rather than tacit knowledge that they may have developed, a naïve coder was also used. Although a high match rate and reliability only demonstrate consistency, not necessarily accuracy, a sufficiently high match rate and reliability give a level of confidence in the taxonomy. Unlike many common applications of inter-rater reliability, in this research the archive itself is also a source of error since entries themselves could be unclear or incomplete. Thus, it is important to distinguish between mismatches due to insufficiencies with the taxonomy or lack of coder training and mismatches due to insufficient information in an archive entry. In this research, an additional code assessing the level of archive information for each entry was used for this purpose.

### 5.1. Coder agreement results

A total of 127 out of the total of 822 action/event entries were coded by both primary coders and then assessed for inter-rater reliability. Only entries where all three codes (type, embodiment, and purpose) matched were considered a match; entries where two of the codes matched but the third did not were considered mismatches.

Of the 127 entries coded by both primary coders, 79 entries had exact matches, for a match rate of 62.2% and a kappa of 0.556. [Table 2](#) summarises the details of the types

of coding match results. The majority (66.7%) of mismatches were archive related. *Insufficient contextual information* refers to entries without enough information to make a clear classification, resulting in individual coders having to make assumptions and taking a 'most likely' guess. Coders using different assumptions may result in different codes being assigned to an entry. *Multiple Actions* refers to an entry containing several different actions. In this case it becomes ambiguous as to which code should be used, hence leading to different codes if the coders used different actions within an entry. *Coder-related* is when the code was assigned erroneously by a coder. *Taxonomy-related* is used for an entry in which the taxonomy did not have a clear definition or where multiple categories may apply, resulting in different codes.

The naïve coder coded 63 of the 127 entries that were coded by both coders. Of those 63 entries, 43 exactly matched the final codes decided upon by the primary coders after comparing their codes, for a match rate of 68.3% and a kappa of 0.628. This value, however, is not directly comparable to the inter-rater reliability between the primary coders above. That value measures how often the two coders matched each other's codes after coding a subset of the data independently, prior to comparing their codes. The value for the naïve coder measures how often the codes matched the final, agreed-upon codes by the primary coders after comparison and discussion.

While coding each entry, whether or not that entry had sufficient information for an accurate code was assessed, prior to comparing if the coders had assigned matching codes. Of the 85 entries where there was sufficient information, there were 63 exact matches, for a match rate of 74.1% and a kappa of 0.687. For the naïve coder on this subset, there were 34 out of 44 exact matches with the final code, for a match rate of 77.3% and a kappa of 0.730. From these results, it was clear that ambiguity in the data contributed significantly to mismatches.

The methodology used in the paper is meant to improve a taxonomy's description of a given archive. If this is the case, fewer mismatches due to the taxonomy should be observed as the taxonomy is refined over multiple iterations and updates. For this research, the number of entries coded between each refinement was usually too few to draw statistical conclusions. However, the taxonomy just prior to splitting *action\_planning* into *approval* and *plan\_or\_schedule* categories was used to code 66 entries. Of those 66 entries, there were 28 exact matches, for a match rate of 42.4%, and a kappa of 0.333. The refinement of the taxonomy improved the match rate to 62.2% and increased kappa from 0.333 to 0.556. The version of the taxonomy used just prior to the final version had a match rate of 59.4% and a kappa of 0.536, similar to the results using the final version, suggesting that refinement was approaching asymptotic levels of improvement and that the process can be considered complete. Table 3 summarises the inter-rater reliability results for the different versions of the taxonomy.

**Table 2.** Coding match results.

Coding match result	Amount	Percentage of entries
Complete match	79	62.2
Insufficient Contextual Information (Archive-related error)	28	22.0
Multiple actions (Archive-related error)	4	3.2
Coder-related error	16	12.6
Taxonomy-related error	0	0.0



## 5.2. Types of classification mismatches

In this study, coding mismatches could be classified into three categories: due to archive, due to taxonomy, or due to coder error. A mismatch was considered to be due to archive if the main cause was insufficient information for a clear assessment of the action or if multiple actions were stated within a single entry, making it difficult to categorise the entry. It was considered due to taxonomy if the cause was unclear language used to describe each category, typically causing an action to be classified in different categories by different coders. Mismatches due to coder error typically were due to a coder not fully understanding contextual information within a risk report.

*Mismatches due to archive.* Two types of mismatches due to archive were encountered. The first type resulted from a lack of contextual clues within the archive entry. This led to coders making different assumptions about the context, resulting in different codes being assigned to the entry. This was the major cause of errors in determining whether an action/event entry should be an action or an event. For example, an action/event entry of 'Operator Preferences Determined' is vague as to whether it was the design organisation performing the determination, in which case it would be an action, or if it was an outside entity performing the determination, and the design organisation was waiting for the results, in which case it would be an event. Also, an entry of 'Program Review' is vague as to the purpose of the review, whether it was to go over already collected data, to review the feasibility of the current planned schedule, or to evaluate the technical parameters and feasibility, each of which belong to a different purpose category.

The second type of mismatch due to archive occurred when an action/event entry contained multiple actions. For example, an entry of 'Install part X' with the success criterion of 'part X tested to match predicted values' has within it both implementation and validation actions. With such an action/event entry, one coder may choose one action, while the other coder may choose another. Though this type of mismatch was less frequent, it occurred often enough to be included as an error category.

*Mismatches due to coder.* A main cause of mismatches due to coder resulted when a coder did not recognise contextual information, when it was present. For example, an action/event entry of 'Gather Lessons Learned' has the context of gathering knowledge of improvements to the design process gained from previous project experience, and so it should be *gather\_data*, but a coder who is not aware of this may give another code like *evaluation* instead. This type of mismatch can also occur when a coder does not notice an important contextual clue within the larger report entry.

Another type of mismatch due to coder resulted when acronyms were used in risk report entries. Although acronyms are likely understood by people familiar with the project, those unfamiliar with the project may recognise that the acronym is an important

**Table 3.** Inter-rater reliability by taxonomy version.

Taxonomy version	Match rate (%)	Cohen's kappa
Early	42.4	0.333
Near final	59.4	0.536
Final	62.2	0.556
Final with archive-related mismatch entries removed	74.1	0.687

contextual clue, but not know how to incorporate the context into the action/event classification.

Some mismatches are due to coder error, but are also partly due to the archive. For example, the word ‘identify’ is used in the archive to mean different things in different contexts; it may mean to gather data, as in ‘identify all previous risk reports related to current risk’; or to do technical analysis, as in ‘identify optimal solution’; or to plan or schedule, as in ‘identify an acceptable test schedule’. Words such as ‘design’, ‘develop’, and ‘review’ have similar issues, and occasionally a coder reads entries too superficially and this resulted in coding mismatches.

*Mismatches due to taxonomy.* Certain actions did not fit neatly into a particular taxonomy definition, but were initially not encountered and so were not yet considered when refining the taxonomy. For example, setting a goal or target may be considered a planning action, so that managers and engineers have a better idea of how to allocate manpower resources, or it may be considered a technical analysis action, because technical analysis is needed to set a realistic goal. Such actions, though rare, may not clearly fit a taxonomy category when first encountered, leading to different codes being assigned. This type of mismatch should decrease as a taxonomy is refined. There were no mismatches due to taxonomy when the final taxonomy was used in the validation study completed for this research.

### 5.3. Archive entry classification using taxonomy

Table 4 summarises the final coding results, using the final taxonomy on all 822 action/event entries by one of the coders. Action purposes comprise each row and embodiments comprise each column, with the body of the table indicating how many of that purpose–embodiment pair were observed in the archive. A total of 77 entries (9.4% of all entries) were classified as events and are not included in the table; thus, the table only includes the 745 entries that were classified as actions.

From Table 4, *Resource\_planning* and *technical\_evaluation* were the most common purposes of actions, accounting for over half (55.3%) of all actions. Unsurprisingly, *typical* was the majority (68.5%) of embodiments, since it denotes no special embodiment was observed. Other than *typical*, *coordinate* was the second-most observed embodiment, indicating that multiple departments were involved in carrying out an action roughly one-fifth (20.3%) of the time. Many of these were in the planning and evaluation stages of a project. Although the majority of purpose categories did not have many special embodiment codes, *approval* was the exception, with a clear majority of *approval* actions requiring

**Table 4.** Archive action totals.

Purpose	Typical	Inform	Coordinate	Request
Gather_Data	44	20	6	10
Resource_Planning	159	13	47	2
Technical_Evaluation	158	2	29	2
Technical_Planning	74	1	33	1
Approval	1	1	31	31
Implementation	47	0	3	1
Validation_and_Testing	27	0	2	0
Other	0	0	0	0

coordination or another department's or external entity's action. *Implementation and validation\_and\_testing* rarely had any embodiment codes associated with them, indicating they were performed for the most part without the need for involvement by other stakeholders.

The lack of entries in the *other* category indicates that all 745 actions were categorised into one of the seven categories defined by the taxonomy.

## 6. Discussion

### 6.1. Value of a mismatch-driven discovery method

Although classification mismatches should ideally not occur, they can give important clues as to areas for improvement. Thus they should not necessarily be considered as problematic, but rather as important for refining a taxonomy.

*Value of mismatches due to archive.* Mismatches due to archive can point to improvements on how project information is recorded. A lack of easily understood contextual clues in archive entries implies that if more care were taken in recording information, a better understanding of the project and actions taken therein could be preserved. This may also give the engineer an opportunity to more clearly consider the intended actions. It should be noted that in the archive used in this study, relevant contextual information such as design rationale and alternatives considered could have been placed in a provided 'comments' section; however, this was rarely used. Improving archive entry practices could lead to lower rates of coding mismatches and better information preservation.

Additionally, during the refinement process, if a large proportion of the mismatches are due to the archive rather than the coders or the taxonomy, then this indicates that the refinement process is close to completion, since the coders and the taxonomy are not introducing significant amounts of error in the descriptive power of the data represented by the archive.

*Value of mismatches due to coder.* Mismatches due to coder are indicative of the need for coder training and coder familiarity with the archive. As a coder becomes more familiar with the taxonomy and with the archive, this type of mismatch should decrease. It should also be noted that one issue with this type of mismatch was if the archive contained words with multiple meanings, such as 'identify'. If the coder did not read closely enough to fully understand the contextual clues around such words, a coding mismatch could occur. This points to the challenges of using automated keyword-match algorithms to codify data in archives such as the one used in this research. Although such keyword-match algorithms would be ideal and less labour intensive than manual coding, the presence of words that have multiple meanings based on the context makes automation difficult (Ahmed, Kim, and Wallace 2007).

*Value of mismatches due to taxonomy.* The procedure described here relies on mismatches in coder classification to improve the taxonomy. Thus, mismatches early on are crucial for driving discussion on limitations of the existing taxonomy and how to improve the taxonomy. Later versions of a taxonomy should have fewer mismatches than earlier versions of a taxonomy. When the number of mismatches due to the taxonomy is low, it is an indication that the taxonomy is relatively complete and can adequately capture entries presented in a particular archive. In this study, taxonomy-related

mismatches were eventually a relatively insignificant number of all mismatches, and none were observed using the final version of the taxonomy, indicating that the refinement of the taxonomy was fairly complete. Although the final version of the taxonomy may be relatively simplistic, its lack of taxonomy-related mismatches, compared with other taxonomies such as the initial one used for this study, demonstrates the taxonomy's ability to classify risk-mitigating actions for this archive.

## **6.2. Generality of taxonomy-discovery technique and developed taxonomy**

The technique described in this research is applicable to all taxonomy creation problems. Although it is applied in this research to a problem in which expert input is unavailable, it is straightforward to apply the approach to refine a taxonomy initially defined through a typical expert-driven process, and consulting experts or using subject area experts as coders may allow for faster convergence, by reducing the amount of coder-related mismatches. Furthermore, the technique makes no assumptions about the structure or content of the archive other than that it contains enough entries to support the iterative process. This is significant as it frees the taxonomy to reflect the actual concepts and relationships that emerge from the information archival process rather than an imposed (and potentially neglected) standard that archivists are supposed to use. Finally, it is straightforward to extend the basic process described here to related problems, such as thesaurus and ontology creation.

Although the taxonomy developed in this research is based on a design project archive from one particular engineering organisation, the concepts identified are general and likely applicable to other organisations. Moreover, this taxonomy can serve as the seed taxonomy for repeating the refinement process on a similar archive from a different organisation. The taxonomy can be used as a basis for deriving relationships between actions taken during a project and the project's eventual outcome in terms of metrics such as budget and schedule, a future goal of this study. Correlating actions with a project's outcome would lead to a better understanding of which actions to take under different project scenarios.

## **6.3. Considerations during coding**

While using the taxonomy to code the archive, there were several types of mismatches that were more related to how to implement the taxonomy on the archive as opposed to the definitions therein. While coding, it is worthwhile to keep these issues in mind to decrease the number of coder-related mismatches.

Scoping. An action can belong to several different purpose categories depending on the scope of its effect, or at which organisational level of the project is the action considered. For example, updating a build plan could be considered a resource planning action at the level of the overall project, but would be an implementation at the level of the plan itself, because it is putting a change into effect. To decrease these sorts of mismatches, the scope of each action was set at the level of the risk report, so that only the effect relative to the objective of that risk report was considered.

Milestoning. Because the archive was primarily a design history archive, some of the action/event entries were not relevant as-is, but were simply markers for completing a

certain task. For example, an engineer may have felt that an action of 'Redesign part' lacked a clear criteria marking when the action was considered complete, and so may have instead written 'Upload redesigned part dimensions' so that there would be a distinct completion criteria. Thus when looking at each entry, the coder needs to consider whether the written text is concerning the action performed that makes progress on mitigating risk, or if the text is just marking the completion of a risk-mitigating action that needs to be inferred. We call this *milestoning* because the engineer recorded the milestone for an action, rather than the action itself.

#### 6.4. Challenges for classifying actions in a design archive

A taxonomy of risk-mitigating actions is ideally complete, in that it covers the spectrum of possible risk-mitigating actions. However, the taxonomy is based on the archive from which it was developed, and limitations of the data within this particular archive can affect the applicability of the taxonomy.

First, because the taxonomy is based on actions recorded in the archive, actions that were considered too informal would likely not appear in the archive and thus are not a part of the taxonomy. This includes informal discussions among engineers and engineers' notebooks, which serve an important purpose in promoting the generation of ideas, but filter out actions that are deemed to be less likely to succeed, and hence never entered into a risk archive (Hertzum and Pejtersen 2000). Because such actions or ideas for actions are not captured within the organisation's formal archive, they may not be captured by the developed taxonomy.

Second, the taxonomy may not encompass certain actions that the archive could record but the author of the entry did not include. An example would be that many risk reports involving a redesign ended with a successful redesign solution, rather than the presumed next step of uploading the solution to internal databases for dissemination. Although the taxonomy can capture these sorts of actions when they are present in a risk report, it is much more difficult to account for their absence, and this could potentially skew the percentage and number of different categories of actions.

#### 6.5. Future work

The development of the taxonomy of risk-mitigating actions discussed in this paper is the basis for further research into extracting useful project knowledge from design project archives. Additional taxonomy discovery will be conducted so that design project events, outcomes, and other project status information can be classified in addition to risk-mitigating actions. This will enable a full encoding of the archive and subsequent investigations into the effectiveness of different actions or sequences of actions in various situations. For example, the archive contains entries regarding whether or not risk-mitigating actions were successful in reducing the risk. This can be correlated with the type of action taken and the type of situation, to determine which actions tended to be more successful in which situations. In this way, the archival information unlocked via the taxonomy-refinement process will allow for more effective actions to be applied in future projects.

## 7. Implications and conclusion

This paper describes a technique for discovering a taxonomy that adequately classifies entries in a textual or weakly structured archive of design information. The technique is general and applies in situations in which domain expertise is insufficient or inaccessible. The discovery process is seeded by an initial taxonomy that can be rooted in the available expertise, and the taxonomy is subsequently refined through an iterative process to better reflect the contents of the archive. A demonstration is presented of generating a taxonomy of risk-mitigating actions using reports from an engineering design organisation's legacy project risk archive. During the final stage of refinement, no mismatches due to the taxonomy were found, demonstrating that the iterative process resulted in a taxonomy that can adequately describe the risk-mitigating actions found in the archive. Use of a naïve coder also validated that the final taxonomy definitions adequately describe the risk-mitigating actions. This taxonomy-discovery process can help recover useful information from legacy archives, and further applications of the taxonomy presented in this paper will improve the understanding of risk-mitigating actions and their effects on projects, allowing for better decision-making in future projects.

## Acknowledgements

The authors gratefully acknowledge the assistance of Logan Rector for this paper. Opinions expressed in this paper are of the authors and do not necessarily reflect the views of the National Science Foundation.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## Funding

This material is based upon work supported by the National Science Foundation [grant numbers CMMI-1029964 and CMMI-1030060].

## ORCID

Chuck Hsiao  <http://orcid.org/0000-0002-6169-9041>

## References

- Adler, M. D., and K. B. Johnson. 2000. "Quantifying the Literature of Computer-Aided Instruction in Medical Education." *Academic Medicine* 75 (10): 1025–1028.
- Ahmed, S., S. Kim, and K. M. Wallace. 2007. "A Methodology for Creating Ontologies for Engineering Design." *ASME Journal of Computing Information Science and Engineering* 7 (2): 132–140.
- Aloini, D., R. Dulmin, and V. Mininno. 2007. "Risk Management in ERP Project Introduction: Review of the Literature." *Information & Management* 44 (6): 547–567.
- Auerbach, C. F., and L. B. Silverstein. 2003. *Qualitative Data: An Introduction to Coding and Analysis*. New York: New York University Press.
- Bakeman, R., D. McArthur, V. Quera, and B. F. Robinson. 1997. "Detecting Sequential Patterns and Determining Their Reliability with Fallible Observers." *Psychological Methods* 2 (4): 357–370.

- Bedford, T., and R. M. Cooke. 2001. *Probabilistic Risk Analysis: Foundations and Methods*. Cambridge: Cambridge University Press.
- Berry, K. J., and P. W. Mielke. 1988. "A Generalization of Cohen's Kappa Agreement Measure to Interval Measurement and Multiple Raters." *Educational and Psychological Measurement* 48 (4): 921–933.
- Boehm, B. W. 1991. "Software Risk Management: Principles and Practices." *Software, IEEE* 8 (1): 32–41.
- Brennan, P. F., and B. J. Hays. 1992. "Focus on Psychometrics: The Kappa Statistic for Establishing Interrater Reliability in the Secondary Analysis of Qualitative Clinical Data." *Research in Nursing & Health* 15 (2): 153–158.
- Cantor, A. B. 1996. "Sample-Size Calculations for Cohen's Kappa." *Psychological Methods* 1 (2): 150–153.
- Carr, M. J., S. L. Konda, I. Monarch, F. C. Ulrich, and C. F. Walker. 1993. *Taxonomy-Based Risk Identification*. Pittsburgh, PA: Carnegie Mellon University, Software Engineering Institute, CMU/SEI-93-TR-6, ADA266992.
- Catic, A., and J. Malmqvist. 2013. "Effective Method for Creating Engineering Checklists." *Journal of Engineering Design* 24 (6): 453–475.
- Ceesay, B., and W. J. Hou. 2015. "NTNU: An Unsupervised Knowledge Approach for Taxonomy Extraction." Paper presented at the proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), Denver, CO, USA, June 4–5.
- Cheung, C. F., W. B. Lee, and Y. Wang. 2005. "A Multi-Facet Taxonomy System with Applications in Unstructured Knowledge Management." *Journal of Knowledge Management* 9 (6): 76–91.
- Cheung, C. F., W. B. Lee, W. M. Wang, Y. Wang, and W. M. Yeung. 2011. "A Multi-Faceted and Automatic Knowledge Elicitation System (Makes) for Managing Unstructured Information." *Expert Systems with Applications* 38: 5245–5258.
- Clark-Carter, D. 1997. *Doing Quantitative Psychological Research: from Design to Report*. Hove: Psychology Press/Erlbaum.
- Cohen, J. 1960. "A Coefficient of Agreement for Nominal Scales." *Educational and Psychological Measurement* 20 (1): 37–46.
- Coughlan, P., and D. Coughlan. 2002. "Action Research for Operations Management." *International Journal of Operations & Production Management* 22 (2): 220–240.
- De Knijff, J., F. Frasinicar, and F. Hogenboom. 2013. "Domain Taxonomy Learning from Text: The Subsumption Method Versus Hierarchical Clustering." *Data & Knowledge Engineering* 83: 54–69.
- Gilchrist, A. 2003. "Thesauri, Taxonomies, and Ontologies – An Etymological Note." *Journal of Documentation* 59 (1): 7–18.
- Greenfield, M. A. 2000. "NASA's use of quantitative risk assessment for safety upgrades." IAA symposium. Rio de Janeiro, Brazil, October 2–6.
- Heisig, P., N. H. M. Caldwell, and P. J. Clarkson. 2014. "Core Information Categories for Engineering Design – Contrasting Empirical Studies with a Review of Integrated Models." *Journal of Engineering Design* 25 (1–3): 88–124.
- Hepp, M. 2008. "Ontologies: State of the Art, Business Potential, and Grand Challenges." In *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*, edited by M. Hepp, P. De Leenheer, A. De Moor and Y. Sure, 3–22. New York City, NY: Springer US.
- Hertzum, M., and A. M. Pejtersen. 2000. "The Information-Seeking Practices of Engineers: Searching for Documents as well as for People." *Information Processing and Management* 36 (5): 761–778.
- Jagtap, S., and A. Johnson. 2011. "In-Service Information Required by Engineering Designers." *Research in Engineering Design* 22: 207–221.
- Kara, S., O. Alan, O. Sabuncu, S. Akpinar, N. K. Cicekli, and F. N. Alpaslan. 2012. "An Ontology-Based Retrieval System Using Semantic Indexing." *Information Systems* 37 (4): 294–305.
- Kurtoglu, T., M. I. Campbell, C. B. Arnold, R. B. Stone, and D. A. Mcadams. 2009. "A Component Taxonomy as a Framework for Computational Design Synthesis." *ASME Journal of Computing and Information Science in Engineering* 9 (1): 011007.
- Landis, J. R., and G. G. Koch. 1977. "The Measurement of Observer Agreement for Categorical Data." *Biometrics* 33 (1): 159–174.



- Li, T., P. Chubak, L. V. S. Lakshmanan, and R. Pottinger. 2012. "Efficient Extraction of Ontologies from Domain Specific Text Corpora." Paper presented at the Proceedings of the 21st ACM international conference on information and knowledge management, Maui, HI, USA, October 29–November 2.
- Li, L., F. Qin, S. Gao, and Y. Liu. 2014. "An Approach for Design Rationale Retrieval Using Ontology-Aided Indexing." *Journal of Engineering Design* 25 (7–9): 259–279.
- Liang, Y., Y. Liu, C. K. Kwong, and W. B. Lee. 2012. "Learning the "Whys": Discovering Design Rationale Using Text Mining – an Algorithm Perspective." *Computer-Aided Design* 44 (10): 916–930.
- Lowe, A., C. McMahon, T. Shah, and S. J. Culley. 2000. "An Analysis of the Content of Technical Information Used by Engineering Designers." Paper presented at the ASME design engineering technical conferences. Baltimore, MD, September 10–13.
- McMahon, C., A. Lowe, S. J. Culley, M. Corderoy, R. Crossland, T. Shah, and D. Stewart. 2004. "Waypoint: An Integrated Search and Retrieval System for Engineering Documents." *ASME Journal of Computing and Information Science in Engineering* 4 (4): 329–338.
- Medelyan, O., I. H. Witten, A. Divoli, and J. Broekstra. 2013. "Automatic Construction of Lexicons, Taxonomies, Ontologies, and Other Knowledge Structures." *Data Mining and Knowledge Discovery* 3 (4): 257–279.
- Miller, R., and D. Lessard. 2001. "Understanding and Managing Risks in Large Engineering Projects." *International Journal of Project Management* 19 (8): 437–443.
- Muehlen, M., and D. Ho. 2006. "Risk Management in the BPM Lifecycle." In *Business Process Management Workshops*, edited by C. Bussler, and A. Haller, 454–466. Berlin/Heidelberg: Springer.
- Paukkeri, M.-S., A. P. Garcia-Plaza, V. Fresno, R. Martinez Unanue, and T. Honkela. 2011. "Learning a Taxonomy from a Set of Text Documents." *Applied Soft Computing* 12 (3): 1138–1148.
- Peltier, T. R. 2004. "Risk Analysis and Risk Management." *EDPACS: The EDP Audit, Control, and Security Newsletter* 32 (3): 1–17.
- Regli, W. C., J. B. Kopena, and M. Grauer. 2011. "On the Long-Term Retention of Geometry-Centric Digital Engineering Artifacts." *Computer-Aided Design* 43 (7): 820–837.
- Robson, C. 1993. *Real World Research: A Resource for Social Scientists and Practitioner-Researchers*. Cambridge, MA: Blackwell.
- Rockwell, J. A., I. R. Grosse, S. Krishnamurty, and J. C. Wileden. 2010. "A Semantic Information Model for Capturing and Communicating Design Decisions." *ASME Journal of Computing and Information Science in Engineering* 10 (3): 031008.
- Salzberg, S., and M. Watkins. 1990. "Managing Information for Concurrent Engineering: Challenges and Barriers." *Research in Engineering Design* 2 (1): 35–52.
- Simperl, E. P. B., T. Burger, S. Hangl, S. Worgl, and I. Popov. 2012. "Ontocom: A Reliable Cost Estimation Method for Ontology Development Projects." *Web Semantics: Science, Services and Agents on the World Wide Web* 16: 1–16.
- Spooren, P., D. Mortelmans, and J. Denekens. 2007. "Student Evaluation of Teaching Quality in Higher Education: Development of an Instrument Based on 10 Likert-Scales." *Assessment & Evaluation in Higher Education* 32 (6): 667–679.
- Stamatelatos, M., and G. Apostolakis. 2002. *Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners v 1.1*. Washington, DC: NASA.
- Stamatis, D. H. 2003. *Failure Mode and Effect Analysis: FMEA from Theory to Execution*. Milwaukee: ASQ Quality Press.
- Thunnissen, D. P. 2003. "Uncertainty Classification for the Design and Development of Complex Systems." Paper presented at the 3rd annual predictive methods conference. Newport Beach, CA, June 16–17.
- Tsui, E., W. M. Wang, L. Cai, C. F. Cheung, and W. B. Lee. 2014. "Knowledge-Based Extraction of Intellectual Capital-Related Information from Unstructured Data." *Expert Systems with Applications* 41: 1315–1325.
- Vermaas, P. E. 2013. "The Coexistence of Engineering Meanings of Function: Four Responses and their Methodological Implications." *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27: 191–202.
- Vesely, W. E., F. F. Goldberg, N. H. Roberts, and D. F. Haasl. 1981. *The Fault Tree Handbook*. Washington, DC: US Nuclear Regulatory Commission.

- Wilcox-Herzog, A. 2004. "Actions Speak Louder than Words: How Experience and Education Relate to Teachers' Behaviors." *Journal of Early Childhood Teacher Education* 25 (1): 11–18.
- Yu, J., J. Cha, and Y. Lu. 2012. "Design Synthesis Approach Based on Process Decomposition to Design Reuse." *Journal of Engineering Design* 23 (7): 526–543.

Copyright of Journal of Engineering Design is the property of Taylor & Francis Ltd and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.