Aggregating Expert-Driven Causal Maps for Web Effort Estimation

Simon Baker and Emilia Mendes

Computer Science Department, The University of Auckland,
Private Bag 92019, Auckland, New Zealand
sbak030@aucklanduni.ac.nz, emilia@cs.auckland.ac.nz

Abstract. Reliable Web effort estimation is one of the cornerstones of good Web project management. Hence the need to fully understand which factors affect a project's outcome and their causal relationships. The aim of this paper is to provide a wider understanding towards the fundamental factors affecting Web effort estimation and their causal relationships via combining six different Web effort estimation causal maps from six independent local Web companies, representing the knowledge elicited from several domain experts. The methodology used to combine these maps extended previous work by adding a mapping scheme to handle complex domains (e.g. effort estimation), and the use of an aggregation process that preserves all the causal relations in the original maps. The resultant map contains 67 factors, and also commonalities amongst Web companies relating to factors and causal relations, thus providing the means to better understand which factors have a causal effect upon Web effort estimation.

Keywords: Web effort estimation, causal maps, Web effort prediction, map aggregation.

1 Introduction

A cornerstone of Web project management is effort estimation, the process by which effort is forecasted and used as basis to predict project costs and allocate resources effectively, so enabling projects to be delivered on time and within budget [1]. Effort estimation is a very complex domain where the relationship between factors is non-deterministic and has an inherently uncertain nature.

There have been numerous previous attempts to model effort estimation of Web projects, but none yielded a complete causal model incorporating all the necessary component parts. Mendes and Counsell [3] were the first to investigate this field by building a model using machine-learning techniques with data from student-based Web projects, and size measures harvested late in the project's life cycle. Mendes and collaborators also carried out a series of consecutive studies (e.g. [4]-[20]) where models were built using multivariate regression and machine-learning techniques and used data on industrial Web projects. Later they proposed and validated size measures harvested early in a project's life cycle, therefore better suited to effort estimation [1]' when compared to other Web effort predictors previously proposed [21]. Reifer [22]

T.-h. Kim et al. (Eds.): ASEA 2010, CCIS 117, pp. 264–282, 2010.

proposed an extension of an existing software engineering resource model, and a single size measure harvested late in the project's life cycle. None were validated empirically. This size measure was later used by Ruhe et al. [23], who further extended a software engineering hybrid estimation technique to Web projects, using a small data set of industrial projects, with this technique mixing expert judgement and multivariate regression. Baresi et al. [24][25] and Mangia et al. [26] investigated effort estimation models and size measures for Web projects based on a specific Web development method. More recently there have been a number of studies describing causal maps for Web effort estimation [27]-[30], where their causal relationships were identified by a domain expert, using only the set of factors that are part of the Tukutuku database [32]. Other more recent studies compared Web effort prediction techniques, based on existing datasets [33]-[36].

There are issues with all previous studies in that none, when identifying important factors for Web effort estimation, focused solely on factors that presented a cause & effect relationship with effort, i.e., they included any factors correlated with effort. In addition, when surveying companies to identify suitable effort predictors, those studies did not assess how good these companies were at estimating effort for their Web projects.

As part of a NZ government-funded research, Mendes elicited several company-specific expert-driven Web effort estimation causal maps from NZ Web companies [26]. The elicitation process employed is detailed in [26]. Each map was part of a larger model, named a Bayesian Network model (detailed in Section 2), and provided a representation of the Web effort estimation domain from the perspective of the single Web company from which that model had been elicited. All participating companies were consulting companies that developed different types of Web applications (e.g. static application, applications that used a content management system, database-driven Web applications).

Experience from eliciting such maps showed that companies found it much easier and more effective to use an initial set of factors as basis to elicit their causal maps, rather than to build one from scratch [29][30]. In addition, anecdotal evidence obtained throughout the elicitation process with these companies revealed that the use of an aggregated causal map representing the expert knowledge of different Web companies with regard to factors and relationships relevant for Web effort estimation would be extremely useful to help with the elicitation of company-specific causal maps. Some Web companies would like to use such large causal map at the start of the elicitation process; others believed that such maps would be extremely useful to provide them with a sort of "checklist" against which to compare their own, once it had been elicited.

In addition, we believe that such an aggregated map is important for practitioners for the following reasons:

- It depicts a blueprint of the most common factors and causal relationships important for companies that develop and manage Web projects as part of their core businesses. Such knowledge can be very useful to Web projects managers to help them revisit the factors they currently take into account during an effort estimation process.
- Provides companies with a starting point to building a single-company causal map, which can later be used to create a single-company Bayesian Network model. Anecdotal evidence shows that when eliciting factors, causal relationships

and even probabilities using as its basis tacit knowledge, companies find it much easier and more effective to customise an existing causal map to their needs, rather than to build one from scratch. This is also an approach suggested in [29][30].

- It can be used as a benchmark model (i.e. 'measuring stick') to compare different single-company Web effort estimation causal maps.

The contribution of this paper is therefore to provide a wider understanding on the fundamental factors affecting Web effort estimation and their causal relationships via combining six different Web effort estimation causal maps, elicited from six independent local Web companies, representing the knowledge elicited from several domain experts.

The remainder of this paper is structured as follows: an introduction to Bayesian Networks is given in Section 2, for those unfamiliar with this model, followed by a discussion relating to the aggregation of different causal maps (Section 3), and a summary of prior work (Section 4). Then we detail our proposed solution (Section 5) and methodology (Section 6) for aggregating causal maps, followed by a discussion of our results in Section 7. Finally the threats to the validity of our approach and our conclusions are given in Sections 8 and 9, respectively.

2 Bayesian Networks

A Bayesian Network (BN) is a probabilistic model that allows for reasoning under uncertainty. A BN is composed of two components [1]. The first is a graphical causal map, depicted by a Directed Acyclic Graph (DAG) (see Figure 1). The DAG's nodes represent the relevant variables (factors) in the domain being modelled, which can be of different types (e.g. observable or latent, categorical). The DAG's arcs represent the causal relationships between variables, where relationships are quantified probabilistically. These graphs may be simple (as in the example in Figure 1), or very complex in terms of nodes and relations.

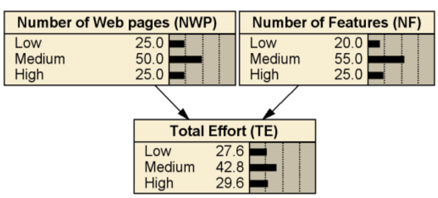

Fig. 1. A small Bayesian Network model and its CPTs

The second component of a BN is the quantitative part: a Conditional Probability Table (CPT) associated to each node in the network. A parent node's CPT describes the relative probability of each state (value); a child node's CPT describes the relative probability of each state conditional on every combination of states of its parents. Each row in a CPT represents a conditional probability distribution and therefore its values sum up to one [1]. A detailed description of the process employed to build BNs is detailed in [38]. The work presented herein is only concerned with the first component of a Bayesian Network, i.e. the causal map graph.

3 Problems Relating to the Aggregation of Causal Maps

It is often recommended that such causal maps be constructed through elicitation from different domain experts in order to derive a comprehensive and accurate causal map [39][40][41][45]. However, it is difficult to combine the beliefs of different experts in a coherent and impartial manner.

In order to arrive at a comprehensive causal map we would need to consult domain experts, many of whom working for different and perhaps competing companies, and thus likely to have a different prospective about the Web development domain. Therefore, the difficulty in combining expert-based causal maps increases for the following reasons:

- Identifying Common Variables: Different experts might represent semantically equivalent concepts in their causal maps using different variable names (e.g. '*Number of Developers*' vs. '*Project Human Resources*'). Furthermore, experts might use a different number of variables to represent the same concept.
- Conflicting Causal Relations: Variables might have contradictory causal relations according to different experts. Two kinds of causal relation conflicts can occur: the first when there is a causal influence between two variables according to an expert's belief, which is strictly prohibited by another expert's belief. The other type of conflict is the occurrence of cycles (which is ruled out within the context of this work in order to keep the resulting aggregated causal map consistent with all the six individual maps being used as input, which were all Directed Acyclic Graphs (DAGs).
- Collaboration Constraints: One feasible way to construct a generic causal map for Web effort estimation is to elicit a single map from a group of domain experts from a representative sample of Web development companies. This would need to be done in stages, and such approach might work well with small groups of domain experts but will likely to be impractical when additional models are included in the unified model. However, within the context of this research, any form of collaboration between domain experts is not feasible because all of the participating companies compete in the same market. This means that, by collaborating with other experts, they would be forced to share sensitive business information that they are not willing to disclose.

Therefore, it is vital to apply a methodology for combining different expert-elicited causal maps that solves the difficulties abovementioned. In this paper we propose a method, detailed in Sections 5 and 6, which solves many of the affiliated challenges in combining expert-elicited BNs that have not been sufficiently addressed in prior work.

Note that although all the causal maps that are used as input to our aggregated causal map are part of larger models - BNs, i.e., all represent the qualitative parts of single-company BNs that have been previously built for Web effort estimation, it is not our aim herein to build a cross-company Bayesian Network model for Web effort estimation, but rather to focus solely on the merging of causal maps.

4 Related Work

There are well established techniques that attempt to find consensus between expert opinions such as the Delphi method [37] and many others [41]. However, these methods generally require collaboration and information sharing, and are time consuming. As detailed in Section 3, such requirements are not suitable within the context of this work.

Sagrado and Moral [42] have proposed a qualitative method that combines separate BN Structures as either a union or intersection of DAGs. They consider the initial BN Structures to be I-Maps; I-Maps can be defined as BNs that contain probability parameters consistent with all Markov assumptions implied by its causal structure [43] . The Markov assumption states that a given node is independent of any non-descendent nodes in the network given its parent nodes. Figure 2 shows an example of the Markov assumption at node *X*. The highlighted nodes, (*D, C*, and *G*) are not influenced by *X* (and likewise, cannot influence *X*) given the probabilities of its parent nodes *A* and *B*. An I-Map is a BN possessing probability parameters that are consistent with the Markov assumption at every node.

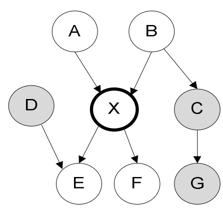

Fig. 2. Markov Assumption at node X

Sagrado and Moral based their approach on a proposition established by Castillo, Gutiérrez, and Hadi [44] , which states: Given two I-Maps defined over the same set of variables, if there is a common ancestral ordering in both I-maps, then the DAG obtained from their intersection is a directed minimal I-Map. In other words, it is possible to derive an intersection structure of two independent causal maps that share the same variables if there is at least one common prior node. If so, then the resultant intersection structure will be a minimal I-Map, which is an I-Map that cannot be simplified any further. As a consequence, the intersection structure will conserve all independencies that exist in the original causal structures.

The main constraint in the proposition by Castillo et al. is that both structures must share exactly the same variables. Sagrado et al.'s approach bypasses this limitation by extending the notion of intersection to "extended intersection", which is essentially a union of the two causal maps in reference to the starting prior node; i.e. all nodes reachable from the starting prior node in both causal structures. This solution suits our situation because of the automation it provides, but more importantly, it is non-intrusive and allows us to combine causal maps without the need to ask unwilling domain experts to disclose business process information to other domain experts. Nevertheless, this approach has never been empirically verified, and the authors present only a theoretical account of this method without any real world example. In terms of our research, the drawbacks to this method are as follows:

It assumes that individual models will share at least one prior node in order to perform the union/intersection operation. Although this might seem very probable for our case, given that all the participating companies share the same problem domain, in reality this is not the case. All the company models have defined variables that are very specific to their business process; even though there exist similar nodes shared between the models, their exact definition and context makes them different.

Sagrado and Moral's approach does not preserve, unlike the original proposal by Castillo et al. [44], independence relationships; in fact, it can result in the contradiction of the topological rules of BN causal structures, such as the introduction of cycles. However, they show that whenever there are no head-to-head (converging) edges, the resultant causal map is a minimal I-Map.

We believe that aggregating structures provides one with much more relevant information than to a simple union or intersection of structures, as we can distinguish between more common factors and causal relations in the domain, whereas a simple union or intersection looses this information in the process [45].

5 Proposed Solution

We propose a qualitative methodology that pragmatically addresses the shortcomings of Sagrado et al.'s approach by:

1. Introducing a mapping scheme, i.e., a way to identify similar existing variables in the participating companies' BN models.
2. Instead of using a simple union/intersection, which can only include a common node or edge exactly once, we attempt to aggregate the causal structures. By aggregation, we imply that all edges and nodes in the original map are preserved. As more causal maps are aggregated, the most common variables and causal links emerge, thereby simulating in our view a form of consensus between the different companies' maps.

We termed the resultant aggregated graph as a Causal Structure Aggregation Model (CSAM). Strictly speaking, it is not a unified causal map, but it is a tool for discovering a consensual causal maps. A CSAM is a graph that represents the cumulative union of individual causal maps according to a node mapping scheme. The aim of a structure aggregation model is to identify causal commonalties between independently developed maps that share the same domain. Consider the three causal maps presented in Figure 3. All are used to estimate the total effort required to develop a Web application. Since they all share the same domain, it is possible to assume that the nodes in two different models portray the same factor. For example, nodes A1, B1, and C1 all model the same factor - *the number of developers required to develop a Web application*, and therefore, it is possible to map those three nodes into a single factor.

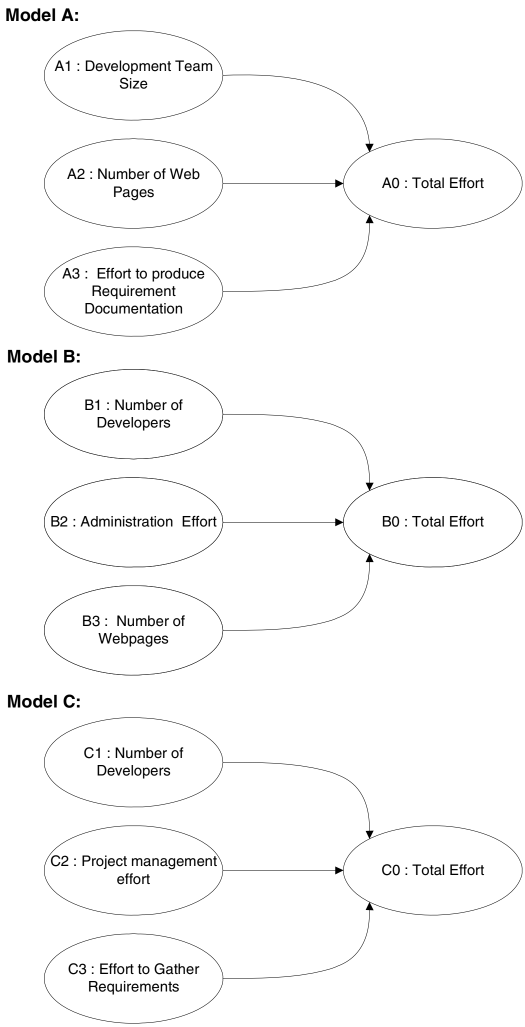

Fig. 3. Three Basic Examples of Causal Maps

Some nodes are more subjective in their definition, e.g., nodes B2 and C2 both attempt to model the effort required to develop a Web application, but the exact details of how to measure this effort might vary between the two companies. However, because both models share the same domain, both B2 and C2 are likely to portray the same underlying concept. By performing this type of mapping between the three models, we can produce the CSAM presented in Figure 4.

The left partition of a CSAM's node represents a factor of interest, while the right partition contains a list of nodes from the original models that map to this factor. All the causal links from the original models are preserved in the CSAM, i.e., if there is a link between two nodes in one of the original models (for example from A2 to A0), then in the CSAM there must be an edge from every node that contains A2 in its mapping to every node that contains A0 in its mapping. The numbers attached to the edges in the CSAM represent the cardinality of their mapping. For example, the edge from node (1) to node (0) has a cardinality of three; this is because there are three original edges that map to it: A1 to A0, B1 to B0, and C1 to C0. The cardinality of a node is the number of 'original' nodes that it maps to (i.e. the number of nodes listed in its right partition). The example in Figure 4 has a simple one-to-one mapping between the CSAM factors and the nodes from the example causal maps. It is possible to have a many-to-many mapping to resolve more ambiguous situations.

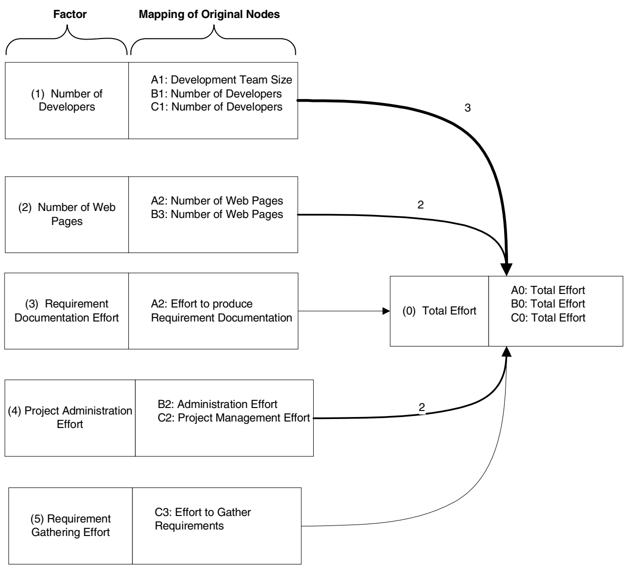

Fig. 4. Causal Structure Aggregation Model (CSAM) of the causal maps in Fig. 3

6 Methodology

The goal of this research was to build a CSAM by aggregating six different expert-driven Web effort estimation causal maps. The methodology used to combine these maps comprised a six-step process (detailed below) combining both linear and iterative approaches (see Figure 5).

1) Formatting of the companies' causal maps

The companies' maps were first formatted so they could be handled by the aggregation algorithm (step 4). The formatting consisted of the following steps:

Each node in every map was given a unique Identifier. The identifiers chosen for our research represented a concatenation between a company's causal map identifier and a unique natural number (a number only valid within the context of a single causal map). Each causal map was represented in a parseable format, where the format chosen herein was CSV (Comma Separated Values).

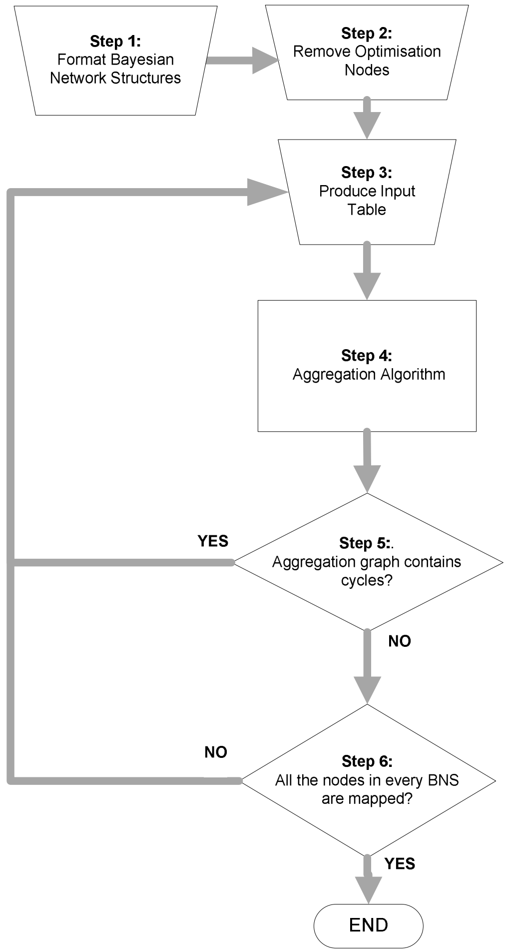

Fig. 5. Process Flow Diagram for Producing a CSAM

The choice relating to the identifiers' representation and parsing format to use was informed by the tool implemented to help with this aggregation process.

2) Removal of Optimisation Nodes

Optimisation nodes are intermediate nodes that were inserted into a causal structure to partition large CPTs in order to reduce their probability elicitation effort. In general, such nodes are not part of the original map elicited with the domain experts; rather, they are suggested by the Knowledge Engineer, and approved by the experts. The purpose of our CSAM was to only aggregate the factors and causal relationships originally modelled by the experts, and as such, the inclusion of optimisation nodes was deemed inappropriate.

Optimisation nodes were first identified from the documentation available for each of the companies' maps. To remove an optimisation node we connected all of its incoming edges (coming from its parent nodes) directly to all of its child nodes, followed by the removal of this optimisation node and all of its outgoing edges (see Fig. 6):

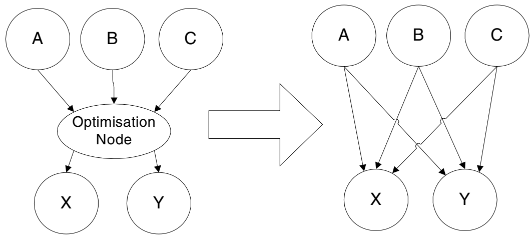

Fig. 6. Removing an Optimization node

During this operation BNs' existing graph rules must always hold (note that each casual map used was part of a larger model – a BN model). For example, only a single edge could have the same source and destination nodes; therefore, if the removal of an optimisation node resulted in adding an edge between two nodes that were already directly linked, then the resultant edge had to be discarded.

3) Creation of an Input Table

Each node in our CSAM corresponded to a semantically equivalent node originating from one of more of the causal maps used as input. Sometimes different causal maps would contain the same node however named differently; when carrying out the mapping (as detailed below) we checked for the semantic equivalence between nodes across causal maps. These mappings were documented using a Table, where each row was used to map a CSAM node to all the other semantically equivalent nodes originating from causal maps. The table's first column represented a CSAM node (factor), identified by a unique ID; the remaining columns contained node identifiers associated with the nodes contained in the input causal maps.

Given a company's input causal map, the first node to be mapped was the most-posterior node, which within our context always happened to be the *Total Effort*. We chose this node because it was part of all the participating companies' causal maps, and therefore we believed it to be the easiest node to identify and map. Once *Total Effort* was mapped, the remaining nodes were mapped according to the following steps:

1. Selection of a node (factor) from a company's causal map that had not yet been mapped.
2. Identification of the contextual meaning of the factor selected in (1), which usually involved interpreting the underlying concept that the DE employed when that factor was elicited. We first identified the units and quantification used to measure the factor, followed by looking at the supporting documentation from the elicitation sessions, which contained examples and additional commentary about the DEs' beliefs. In the rare cases where a factor's contextual meaning was still ambiguous, the DEs were contacted for clarification.

3. Attempt to map the factor identified in (1) (f) to a factor, or set of factors, already present in our CSAM. Whenever there was no corresponding factor(s) clearly mapping to f, we created a new factor(s) within our CSAM to match that given factor f.

There were no strict rules as to whether an original node was mapped to one or more factors within our CSAM; however, we always aimed to keep as much of the original context as possible through the mapping. Thus the reason why our methodology is iterative and not linear is because mappings often change as new factors are created and old ones are revised.

In order to minimise the effort of constantly changing the mappings as the aggregation map was populated, we decided to map the original nodes in different iterations rather than mapping all nodes at once. This gave us the opportunity to run the aggregation algorithm (see step 4 below) and generate the CSAM several times, containing incomplete aggregation maps, and then to look for faults and inconsistencies (e.g. cycles). The first iteration involved mapping every prior node from all the companies' causal maps. The second iteration involved mapping all the nodes from all the companies' maps that were directly pointed to by all prior nodes, and so on until the most posterior (the *Total Effort* Node) was reached.

4) Aggregation Algorithm

The Table prepared in step 3 was used as input to an aggregation algorithm that produced a graphical representation of the CSAM. The algorithm worked by first merging the prior nodes according to the mapping specified in the Table, and continuing until all nodes in all the companies' maps were processed [39].

Whenever the Table from Step 3 did not include mappings for some of the nodes in the inputted causal maps, then these nodes were represented in the CSAM by *placeholder* nodes. The purpose of the placeholder node was so that we were aware of which nodes still required mapping in the next iteration of this process (see Step 6).

5) Check if the Aggregation Graph Contains Cycles

The aggregation algorithm allowed for the occurrence of cycles since it simply followed what was documented in the Table used as input. Therefore, when the generated CSAM graph contained cycles, the input Table needed to be modified so that all of the documented cycles were broken. Cycles could be broken by changing the mapping of one or more nodes that made up the cycle, which could be achieved by either removing or adding factors to the input Table. However, because all the companies' maps were independent of one another and yet shared the same domain, it is theoretically possible, in theory, to have cycles occurring that may not be resolved. This would occur whenever nodes in their original causal maps did not form cycles, but ended up contributing to a cycle in the CSAM due to conflicting contexts.

6) Check if All Nodes are Mapped

The final step in the process was to check whether every node (except for optimisation nodes) in all the companies' causal maps had been mapped in the CSAM. For this we looked for the existence of placeholder nodes in the CSAM outputted by the algorithm. If found, we mapped the map's nodes identified by the placeholder nodes by referring back to Step 3; conversely, if there were no placeholder nodes, we considered that the CSAM was complete according to our mapping.

7 Results

In this section, we present our results from aggregating six independently elicited singly-company causal maps. The elicited models varied in their sizes; as summarised in the following table:

Table 1. List of Causal maps and their sizes

Map	Number of Nodes	Number of Edges
Model A	30	32
Model B	16	20
Model C	15	14
Model D	26	27
Model E	26	29
Model F	19	18

The CSAM[1] resulting from our 6-step methodology (presented in Section 5) enabled us to identify common factors and causal relations shared amongst the six independent single-company BNs. This CSAM presented 67 nodes in total, encompassing all the factors identified by all six participating companies via their BNs. This combined list of factors brought us one step closer to determining all the factors in our target domain (Web development effort estimation), and therefore closer to a unified BN for Web effort estimation. Table 2 lists the Factors[2] in our CSAM and their cardinality, which corresponds to the number of input causal maps that contained that factor. Therefore, a factor's cardinality is an indication of how common this factor was as a predictor amongst the six participating companies.

The most common factor in our CSAM, presenting the highest cardinality on the list, was the '*Number of New Web Pages*'. This may perhaps be an expected result, as the number of Web pages is often likely to be used to determine project scope [2]. This result provides even stronger supporting evidence that there is a causal relationship between new Web pages developed and total development effort; however, given that the sample used was not random we cannot generalise this trend to all remaining Web companies. Another observation is that factors that perhaps may have higher importance in conventional software development did not seem to be common amongst the participating companies; for example, factors related to requirements engineering and documentation. This is perhaps indicative of more agile software development methodologies being employed by the participating companies.

---

[1] The resultant CSAM is available here:
`http://www.cs.auckland.ac.nz/~emilia/ASEA/CSAM.pdf`

[2] A description of all CSAM factors is given here:
`http://www.cs.auckland.ac.nz/~emilia/ASEA/factors.pdf`

Table 2. List of CSAM Factors and their cardinality

Factor Description	Cardinality	Factor Description	Cardinality
Number of new web pages	6	Effort producing animations using software	1
Number of reused web pages	5	Effort programming animations	1
Number of features off the shelf	4	Is development process documented?	1
Project management effort	4	Means of supplying multimedia	1
Adaptation effort of features off the shelf	3	Number of features off the shelf adapted	1
Average project team experience with technology	3	Number of images (new and reused)	1
Development effort of new features	3	Number of images per page	1
Client difficulty	2	Number of images requiring high effort to manipulate	1
Development team size	2	Number of images requiring low effort to manipulate	1
Number of features requiring high effort to create	2	Number of images requiring medium effort to manipulate	1
Number of features requiring high effort to modify/adapt	2	Number of supplied multimedia	1
Number of features requiring low effort to create	2	Pre-development documentation	1
Number of features requiring low effort to modify/adapt	2	Requirements scope	1
Number of images reused	2	Server side language/framework	1
Number of new images	2	Subcontractor development team size	1
Number of web page templates	2	Type images preparation	1
Requirements clarity	2	Type of application	1
Template structure	2	Effort required to train clients	1
Type of project	2	Amount of text per application	1
Client web literacy	2	Quality of third party deliverables	1
Effort template look & feel	2	Number of key client's people	1
Effort to produce web pages	2	Web company's hosting control	1
Effort to produce template mock-up	2	Effort to reuse features off-the-shelf	1
Amount of text per page	1	Effort to adapt features off-the-shelf	1
Are metrics used throughout the project?	1	Effort of graphical design	1
Are the look & feel requirements provided by the client?	1	Effort research third party features	1
Average project team experience (excluding technology)	1	Effort of search engine optimization	1
Average subcontractor development team experience	1	Effort to produce requirements documentation	1
Client location	1	Effort to produce development documentation	1
Client professionalism	1	Effort to develop user interface	1
Technology (client side) language	1	Requirements complexity	1
Data persistence type	1	Effort to program features	1
Development process model	1	Effort to implement the web application	1
Effort spent on images manipulation	1		

Figure 7 shows the proportion of factors according to their equivalent nodal cardinality. We can see that approximately 66% of all factors appeared in only a single company's map, and 34% of factors were common to at least two maps. The percentage of nodes decreased as the cardinality increased, suggesting that the total number of factors available in the target domain significantly outnumbered the factors being considered by individual companies. Likewise, the percentage of causal edges also rapidly decreases with respect to edge cardinality, which suggests that there are many causal relationships not considered by individual companies.

There were 101 causal edges our CSAM. We were able to determine the most common causal relations by selecting all the matched causal relations in the CSAM, that is, all causal edges with a cardinality of two or more. Our results showed that 16% of all causal relations were shared between at least two maps. The most prevalent causal relationship was between factors '*Project Management Effort*' and '*Total Effort*' whereby 66.6% of the participating companies included such relationships in their causal maps. Other common causal relationships identified were:

- Relationship from *Number of New Web Pages*' directly influencing *'Total Effort'*.
- Relationship from *'Number of Reused Web Pages*' directly influencing *'Total Effort'*.
- Relationship from *'Average Project Team Experience with Technology'* directly influencing *'Total Effort'*.

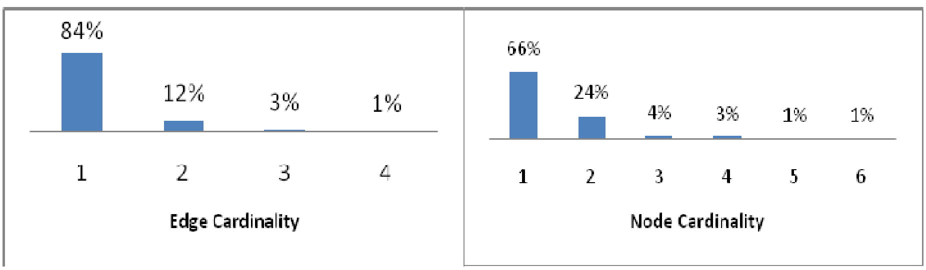

Fig. 7. Distribution of node and edge cardinalities in our CSAM

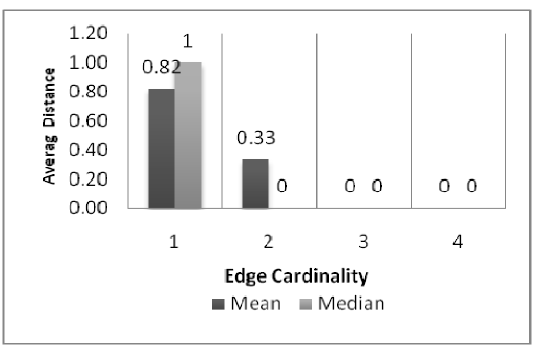

Fig. 8. Average Distance by edge cardinality to Total Effort node

Each of the three abovementioned causal relations appeared in 50% of the companies' maps. Edges with higher cardinality tended to be closer to the most posterior node (*'Total Effort'*). Figure 8 shows a falling trend in the mean and median average distances to the Total Effort node. An average mean distance = 0.82 for edges with cardinality of 1, and mean distance = 0.33 for edges with cardinality of 2. This is in our view an important outcome because 'effort' is what all the causal maps used in this research aim to predict; it is therefore advantageous to know which factors were likely to have a direct effect upon effort, since this would be the focal point of any future consensus-based causal map.

Figure 9 shows a sub-graph of the resultant CSAM with all edges having cardinality greater than two. Factors were grouped into higher level categories (grey boxes) in order to aid readers understand it. This figure can be described as an aggregated intersection of all the causal edges in the inputted causal maps. The higher the weight value of an edge the more common the causal relation is. This figure is therefore very useful as it indicates likely relationships that exist between factors within the Web development domain. We believe that as we further aggregate causal maps to our resultant CSAM, a more informative and decisive consensus will emerge, thus also strengthening the external validity of this model. In other words, a CSAM is a maturing model, providing further certainty as further causal maps are aggregated.

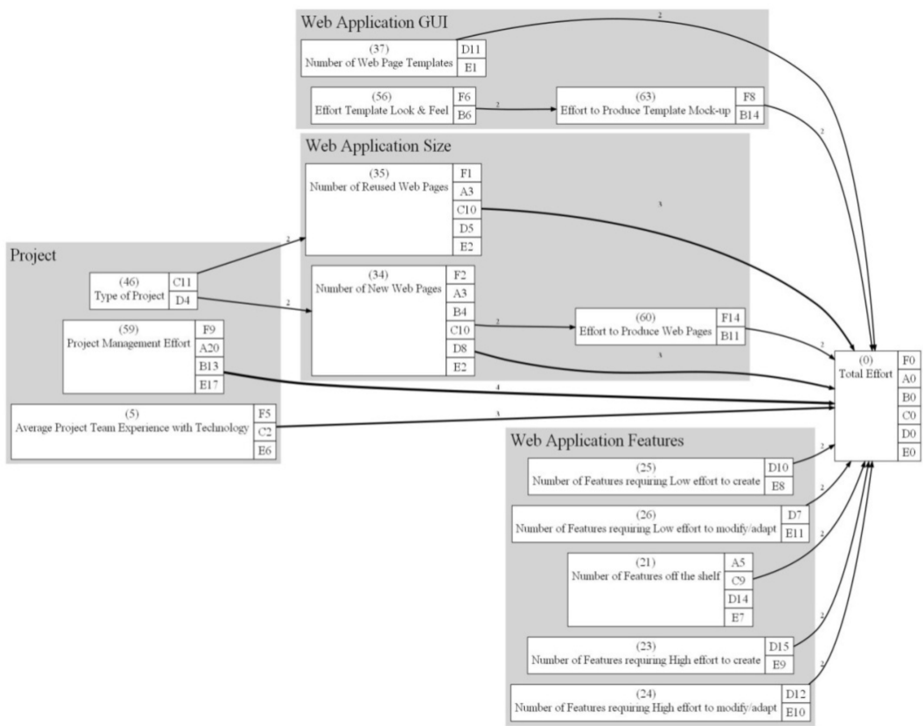

Fig. 9. A sub-graph of the resultant CSAM with all edges having cardinality greater than two (grouped by category)

8 Threats to Validity

There are a few threats to the validity of our work. One is the mapping of original nodes (i.e. creating the aggregation map in third step in our methodology from Section 5). The mapping was performed by the researchers (i.e. knowledge engineers), not the domain experts; therefore, there is always the possibility of bias being introduced. However, it is important to note that many steps were undertaken to mitigate this risk. All mappings were based on documentation provided by the experts, and for cases where there was ambiguity, the experts were contacted directly for further clarification.

Another threat is that our methodology does not in any way guarantee that the final CSAM is free of cycles. Although for the six company maps, all potential cycles were resolved by further investigation and remapping; this might not always be the case. It is always possible to have intrinsically contradictory causal maps, rendering it impossible to resolve cycles unless at least one edge is omitted from the CSAM.

Finally, for the CSAM to be fully comprehensive in terms of domain factors, it is necessary to aggregate a large number of maps. For our case, the aggregation of six maps is not enough to represent all factors and causal relations that impact effort estimation in the Web development domain. However, we note that the resultant CSAM is a maturing model, and we plan to aggregate further causal maps as part of our future work.

9 Conclusions

The aim of this paper was to investigate further the important factors for Web effort estimation and their cause and effect relationships by aggregated six single-company Web effort estimation causal maps. To build such an aggregated model presents numerous challenges, namely identifying common variables, resolving causal relation conflicts, and company collaboration constraints.

We believe that one can overcome some of these challenges by applying an aggregation process that can yield the most common patterns shared between single-company causal maps. Our proposal for building an aggregated map was based on an earlier proposition by Sagrado et al. [42], which attempted to combine BNs' causal maps using intersection/union of DAGs forming a consensus causal structure. Our proposal improved upon this proposition in two ways: first by introducing a mapping mechanism for grouping related variables from different single-company maps, and secondly by using an aggregation of nodes and edges instead of a simple union/intersection, thus preserving all edges and nodes from the original maps. We termed the aggregated causal map as a Causal Structure Aggregation Model (CSAM), and its chief rationale was to identify structural commonalities (common factors and causal relations) found in the original causal maps.

We have constructed a CSAM using six expert-driven single-company causal maps (part of single-company BNs), all of which elicited from local Web development companies in Auckland, NZ. This CSAM contained 67 factors and 101 causal edges. The resultant CSAM revealed the following patterns: i) 34% of the CSAM factors were shared between at least 2 single-company maps; ii) The most common factor was *'Number of New Web Pages'*; a size measure of Web applications, which supports the findings from earlier studies [2]; iii) The proportion of nodes rapidly decreased as cardinality increased, implying that the total number of factors relevant in the Web effort estimation domain significantly outnumbers the number of factors being considered by individual companies; iv) 16% of all causal relations found in the CSAM were shared between at least two single-company maps; v) The most common causal relationship in our CSAM was between factors '*Project Management Effort*' and '*Total Effort*', included in 66.6% of the single-company maps; vi) Three other common causal relationships which were evident were: '*Number of New Web Pages*', '*Number of Reused Web Pages*', and '*Average Project Team Experience with Technology''*, all of which directly influenced '*Total Effort*'; vii) Edges with higher cardinality tended to be closer to the most posterior node, suggesting that most factors influenced total effort directly.

The abovementioned points show that even with a small number of companies we can already see reasonable commonality in terms of factors and causality. The CSAM is a maturing model, which means that as more causal maps are aggregated; the more common factors and causal links will emerge, hence providing an improved consensus. The aggregation process presented herein can be used to aggregate other causal maps. In addition, to our knowledge this is the first time that a study in either Web or Software Engineering describes the creation of a large causal map for effort estimation via the aggregation of several single-company causal maps. Our future work involves the aggregation of other expert-driven single-company causal maps from companies in NZ, and also overseas.

Acknowledgements

We thank Dr. Mosley[*] and Professor Jorgensen for their comments on an earlier version of this paper, and also all the participating companies in this research. This work was sponsored by the Royal Society of New Zealand (Marsden research grant 06-UOA-201).

References

1. Jensen, F.V., Nielsen, T.D.: Bayesian networks and decision graphs. Springer, Heidelberg (2007)
2. Mendes, E., Mosley, N., Counsell, S.: Investigating Web Size Metrics for Early Web Cost Estimation. Journal of Systems and Software 77, 157–172 (2005)
3. Mendes, E., Counsell, S.: Web Development Effort Estimation using Analogy. In: Proc. 2000 Australian Software Engineering Conference, pp. 203–212 (2000)
4. Fewster, R., Mendes, E.: Empirical Evaluation and Prediction of Web Applications' Development Effort. In: Proc. EASE 2000 (2000)
5. Fewster, R., Mendes, E.: Measurement, Prediction and Risk Analysis for Web Applications. In: Proceedings of IEEE Metrics Symposium, pp. 338–348 (2001)
6. Mendes, E., Kitchenham, B.A.: Further Comparison of Cross-company and Within-company Effort Estimation Models for Web Applications. In: Proc. IEEE Metrics, pp. 348–357 (2004)

[*] http://www.metriq.biz

7. Mendes, E., Mosley, N.: Does the Linear Size Adjustment to Estimated Effort Improve Web Applications Effort Estimation Accuracy? Special Issue of the Journal of Computational Methods in Science and Engineering 5(1), 171–184 (2005)
8. Mendes, E., Mosley, N.: Web Cost Estimation: principles and applications. In: Khosrow-Pour, M., Travers, J. (eds.) Web Engineering – Principles and Techniques, pp. 182–202. Idea Group, Inc., USA (2005)
9. Mendes, E., Mosley, N.: Further Investigation into the Use of CBR and Stepwise Regression to Predict Development Effort for Web Hypermedia Applications. In: Proc. ACM/IEEE ISESE, Nara, Japan, pp. 79–90 (2002)
10. Mendes, E., Counsell, S., Mosley, N.: Web Hypermedia Cost Estimation: further assessment and comparison of cost estimation modelling techniques. NRHM 8, 199–229 (2002)
11. Mendes, E., Counsell, S., Mosley, N.: Towards the Prediction of Development Effort for Hypermedia Applications. In: Proc. Hypertext 2001, pp. 249–258 (2001)
12. Mendes, E., Mosley, N., Counsell, S.: Exploring case-based reasoning for Web hypermedia project cost estimation. IJWET 2(1), 117–143 (2005)
13. Mendes, E., Mosley, N., Counsell, S.: A Replicated Assessment of the Use of Adaptation Rules to Improve Web Cost Estimation. In: Proc. ISESE, pp. 100–109 (2003)
14. Mendes, E., Mosley, N., Counsell, S.: Early Web Size Measures and Effort Prediction for Web Costimation. In: Proceedings of the IEEE Metrics Symposium, pp. 18–29 (2003)
15. Mendes, E., Mosley, N., Counsell, S.: Comparison of Length, complexity and functionality as size measures for predicting Web design and authoring effort. IEE Proc. Software 149(3), 86–92 (2002)
16. Mendes, E., Mosley, N., Counsell, S.: The Application of Case-Based Reasoning to Early Web Project Cost Estimation. In: Proc. Compsac 2002, pp. 393–398 (2002)
17. Mendes, E., Mosley, N., Counsell, S.: Web metrics - Metrics for estimating effort to design and author Web applications. IEEE MultiMedia, 50–57 (January-March 2001)
18. Mendes, E., Mosley, N., Counsell, S.: Using an Engineering Approach to Understanding and Predicting Web authoring and Design. In: Proc. HICSC (2001)
19. Mendes, E., Mosley, N., Watson, I.: A Comparison of Case-Based reasoning Approaches to Web Hypermedia Project Cost Estimation. In: Proc. WWW 2002 (2002)
20. Mendes, E., Watson, I., Triggs, C., Mosley, N., Counsell, S.: A Comparative Study of Cost Estimation Models for Web Hypermedia Applications. ESE 8(2), 163–196 (2003)
21. Di Martino, S., Ferrucci, F., Gravino, C., Mendes, E.: Comparing Size Measures for Predicting Web Application Development Effort: A Case Study. In: Proceedings ESEM 2007 (2007)
22. Reifer, D.J.: Web Development: Estimating Quick-to-Market Software. IEEE Software, 57–64 (November-December 2000)
23. Ruhe, M., Jeffery, R., Wieczorek, I.: Cost estimation for Web applications. In: Proceedings ICSE 2003, pp. 285–294 (2003)
24. Baresi, L., Morasca, S., Paolini, P.: An empirical study on the design effort for Web applications. In: Proceedings of WISE 2002, pp. 345–354 (2002)
25. Baresi, L., Morasca, S., Paolini, P.: Estimating the design effort for Web applications. In: Proceedings of Metrics 2003, pp. 62–72 (2003)
26. Mangia, L., Paiano, R.: MMWA: A Software Sizing Model for Web Applications. In: Proc. Fourth International Conference on Web Information Systems Engineering, pp. 53–63 (2003)
27. Mendes, E.: Predicting Web Development Effort Using a Bayesian Network. In: Proceedings of EASE 2007, pp. 83–93 (2007)

28. Mendes, E.: The Use of a Bayesian Network for Web Effort Estimation. In: Baresi, L., Fraternali, P., Houben, G.-J. (eds.) ICWE 2007. LNCS, vol. 4607, pp. 90–104. Springer, Heidelberg (2007)
29. Mendes, E., Polino, C., Mosley, N.: Building an Expert-based Web Effort Estimation Model using Bayesian Networks. In: 13th International Conference on Evaluation & Assessment in Software Engineering (2009)
30. Rajabally, E., Sen, P., Whittle, S., Dalton, J.: Aids to Bayesian belief network construction. In: Proceedings of 2004 2nd International IEEE Conference on Intelligent Systems, vol. 2, pp. 457–461 (2004)
31. Mendes, E., Mosley, N.: Bayesian Network Models for Web Effort Prediction: A Comparative Study. IEEE Trans. on Soft. Engineering 34(6), 723–737 (2008)
32. Mendes, E., Mosley, N., Counsell, S.: Investigating Web Size Metrics for Early Web Cost Estimation. Jour. of Systems and Software 77(2), 157–172 (2005)
33. Corazza, A., Di Martino, S., Ferrucci, F., Gravino, C., Mendes, E.: Applying Support Vector Regression for Web Effort Estimation using a Cross-Company Dataset. In: Proceedings of the ACM/IEEE Symposium on Empirical Software Measurement and Metrics, pp. 191–202 (2009)
34. Corazza, A., Di Martino, S., Ferrucci, F., Gravino, C., Mendes, E.: Using Support Vector Regression for Web Development Effort Estimation. In: Abran, A., Braungarten, R., Dumke, R.R., Cuadrado-Gallego, J.J., Brunekreef, J. (eds.) IWSM 2009. LNCS, vol. 5891, pp. 255–271. Springer, Heidelberg (2009)
35. Ferrucci, F., Gravino, C., Oliveto, R., Sarro, F., Mendes, E.: Investigating Tabu Search for Web Effort Estimation. In: Proceedings of Euromicro SEAA 2010 Conference (2010)
36. Corazza, A., Di Martino, S., Ferrucci, F., Gravino, C., Sarro, F., Mendes, E.: How the Choice of the Fitness Function Impacts on the Use of Genetic Programming for Software Development Effort Estimation? In: Proceedings of PROMISE 2010 (2010) (best paper award)
37. Brown, B.B.: Delphi process: A methodology used for the elicitation of opinions of experts, Santa Monica, CA, Rand Corporation (1968)
38. Woodberry, O., Nicholson, A., Korb, K., Pollino, C.: Parameterising Bayesian Networks. In: Australian Conference on Artificial Intelligence, pp. 1101–1107 (2004)
39. Baker, S.: Towards the Construction of Large Bayesian Networks for Web Cost Estimation. In: Department of Computer Science Auckland: University of Auckland (2009)
40. Montironi, R., Whimster, W.F., Collan, Y., Hamilton, P.W., Thompson, D., Bartels, P.H.: How to develop and use a Bayesian Belief Network. Journal of Clinical Pathology 49, 194 (1996); Mendes, E.: A Comparison of Techniques for Web Effort Estimation. In: First International Symposium on Empirical Software Engineering and Measurement, ESEM 2007, pp. 334–343 (2007)
41. Fink, A., Kosecoff, J., Chassin, M., Brook, R.H.: Consensus methods: characteristics and guidelines for use. American Journal of Public Health 74, 979 (1984)
42. Sagrado, J.D., Moral, S.: Qualitative combination of bayesian networks. International Journal of Intelligent Systems, 237–249 (2003)
43. Flesch, I., Lucas, P., Gamez, J.A., Salmeron, A.: Markov Equivalence in Bayesian Networks (2007)
44. Castillo, E., Gutiérrez, J.M., Hadi, A.S.: Combining multiple directed graphical representations into a single probabilistic model. In: Actas de la Séptima Conferencia Espanola para la Inteligencia Artificial, CAEPIA, pp. 645–652 (1997)
45. Hu, X.-x., Wang, H., Wang, S.: Using Expert's Knowledge to Build Bayesian Networks. In: Proceedings of the 2007 International Conference on Computational Intelligence and Security Workshops, pp. 220–223 (2007)