Zeitschrift: IABSE reports = Rapports AIPC = IVBH Berichte
Band: 68 (1993)

Artikel: Representing incomplete knowledge with assumptions

Autor: Smith, Ian / Boulanger, S.

DOI: https://doi.org/10.5169/seals-51842

Representing Incomplete Knowledge with Assumptions

Représentation de connaissances incomplètes avec hypothèses

Darstellung unvollständigen Wissens mit Annahmen

Ian SMITH
Adj. Dir.AI Lab.
Swiss Fed. Inst.of Technol.
Lausanne, Switzerland

S. BOULANGER
Research Assistant
Swiss Fed. Inst.of Technol.
Lausanne, Switzerland

Ian Smith received a Civil Eng. degree from the Univ. of Waterloo, Canada, and a PhD from Cambridge Univ., UK, in 1982. In 1988, he started the knowledge systems group at ICOM and in 1991, he joined the Artificial Intelligence Lab.

Sylvie Boulanger obtained a B.Sc. in Civil Eng. from the Univ. of Alberta, Canada, and a M.S. from the Univ. of California, Berkeley, USA. Employed for over 5 years at CISC, she is now with the knowledge systems group at ICOM.

SUMMARY

This paper outlines a knowledge representation that accommodates two types of assumptions, defaul and preferences. Knowledge is categorised into three types, models-knowledge known before the ei gineering task begin, rules-knowledge which depends upon decisions made during completion of task, and control knowledge - knowledge related to task planning and conflict resolution, includin knowledge of when to introduce rule clusters, called knowledge islands. Assumptions are accommc dated within models and rules. When used with iterative reasoning strategies, such a representation ha much potential for supporting engineers during the preliminary stages of engineering tasks.

RÉSUMÉ

La représentation des connaissances de l'ingénieur peut être subdivisée en trois catégories: les "Modèles" en tant que personnification du savoir préalable à une tâche, les "Règles" en tant que capacité de décision au cours de l'exécution d'une tâche, et les "Connaissances des commandes" lors de la planification et de la maîtrise des conflits. Il faut y inclure la capacité de discerner le moment voulu pour appliquer les amas de règles (encore désignés par îlots de connaissances scientifiques). Les catégories de modèles et de règles de ce type de représentation s'appuient sur deux hypothèses, à savoir les normes techniques et les préférences personnelles. En liaison avec des stratégies itératives, cette forme de traitement des connaissances est à même de fournir un soutien énorme aux ingénieurs au cours de leurs travaux d'avant-projets.

ZUSAMMENFASSUNG

Die Ergänzung lückenhafter Information erfordert ein umfangreiches Ingenieurwissen. Zur Darstellung wird es in drei Kategorien eingeteilt: "Modelle" als Verkörperung der Vorkenntnisse, "Regeln" als Entscheidungswissen während der Bearbeitung einer Aufgabe, und "Steuerungswissen" bei der Planung und Konfliktbewältigung. Darin enthalten ist die Urteilskraft, wann Regel-Cluster (sog. Wissensinseln) einzusetzen sind. In die Kategorien der Modelle und Regeln sind zwei Typen von Annahmen einbezogen: Fachliche Standards und persönliche Präferenzen. Im Verein mit iterativen Strategien zum Ziehen von Schlüssen verspricht diese Form der Wissensaufbereitung eine beträchtliche Unterstützung der Ingenieurarbeit im Vorprojektstadium.

1 Introduction

There are very few tasks in civil engineering that are accomplished using only complete and exact information. Indeed, the ability to make a good decision in situations of incomplete knowledge is one of the most valuable skills of an experienced engineer. Consequently, knowledge systems which explicitly model strategies for accommodating incomplete knowledge have much potential for supporting civil-engineering tasks.

Engineers traditionally cope with inexact and incomplete information through use of statistical methods. More recently, methods such as default reasoning, fuzzy logic and neural networks have been proposed in order to provide additional strategies for treating imperfect information. These methods are most useful at different stages in the evolution of civil-engineering projects. Thus, they are not necessarily competing strategies; it is, however, important to employ methods that are most suited to the project stage under consideration.

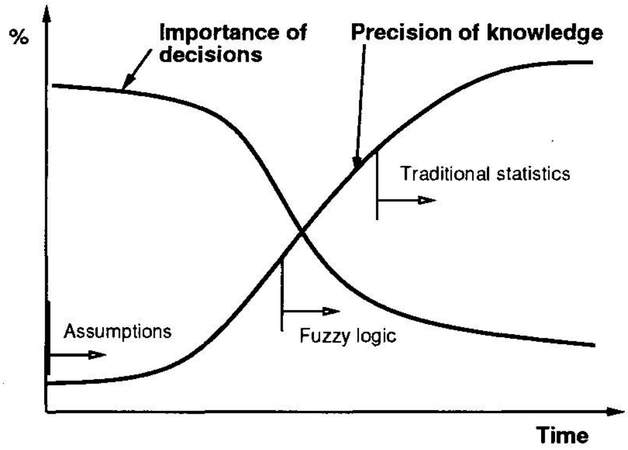

Figure 1: *The importance of design decisions and the precision of knowledge versus time.*

Two characteristic engineering relationships with time are described in Figure 1. The first relationship, *importance of decisions*, describes a trend which decreases slowly at first and then more rapidly until a minimum level is reached. This means that in any engineering task, the most important decisions are made at the beginning. Early decisions have the most influence upon factors such as costs, safety and environmental impact. The second relationship, *precision of knowledge*, describes an opposite trend. Precision (and accuracy) of knowledge is poor at first and increases as tasks near completion. Thus, Figure 1 illustrates a characteristic often used to define engineering practice; important decisions must be made in situations where knowledge is not reliable.

Also shown are positions on this curve where methods that treat imprecision become relevant for implementation. Traditional statistical methods typically require knowledge of all important parameters as well as the shape and size of their distribution. For example, codes recommend relationships derived from these methods. Fuzzy-logic methods (e.g. [1]) rely upon weight factors that usually represent the importance of parameters. Although less information than for traditional statistics is required to use these methods, information such as weight-factor distributions for each parameter is necessary. Therefore, such methods cannot be used at the beginning of engineering tasks. Various neural network methods (e.g. [2]) are proposed for use from the beginning of engineering tasks. However, their usefulness is dependent upon the quality of examples that were used to train the network. Furthermore, it is difficult to maintain performance when these methods are integrated within iterative and decision-making processes as the task proceeds.

The knowledge representation described in this paper results from a study that focuses on supporting decision making during the left portion of Figure 1. A framework for explicit representation of assumptions is presented, and it is shown how this framework can be integrated into a general framework for representation of knowledge to be used during preliminary stages of engineering tasks.

Also, an algorithm which accepts knowledge expressed according to this representation is described. Finally, an example in bridge design is used to illustrate an application in civil-engineering.

2 Representing assumptions

Assumptions are ubiquitous in civil engineering. Perhaps the only situation where civil engineers are able to perform tasks without making assumptions is found during their education, when introductory textbook exercises are completed and simple laboratory experiments are performed. Two types of assumptions are proposed, default and preference. Default information includes all information that is known to be "usually" the case. A great deal of common sense knowledge is default information under this definition. For example, the following statements :

- "Buildings normally have four exterior walls"
- "In this region, piles are not needed for low rise office buildings"
- "Bridges usually have sidewalks on both sides"

contain default information. This type of information is used to make decisions until more information is known.

The second type of assumption, a preference, includes all desired decisions. Therefore, this information reflects a wish to proceed a certain way. If too many problems arise from adopting a preference assumption, the assumption is weakened. Examples of preference assumptions are :

- "Given a choice, study the cheapest alternative"
- "If slope is unstable then avoid placement of bridge piers on the slope"
- "If there is a high risk of earthquakes then continuous bridge spans is a good design for span types"

There are two important differences between defaults and preferences. Firstly, defaults reflect a rough statistical understanding of previous experience in the field. Such an understanding may have no relation to any basic principle within the domain. Preferences, however, reflect the existence of knowledge which is hard to represent more explicitly. Thus, a preference has some relationship to a basic principle which influences the domain. Under this definition, a great deal of civil engineering knowledge can be expressed in preferences since many decisions are influenced by factors that are difficult to model. Such factors include environmental considerations, politics, economic conditions and aesthetics.

The second difference between defaults and preferences is found in their behaviour during situations of conflict with new information. When engineers encounter specific information that conflicts with a default assumption, they simply drop this assumption. Since a default assumption is based solely on empirical observation, there is no reason to do otherwise. However, when a preference assumption is in conflict with more relevant information, it is not dropped completely. Engineers "weaken" the assumption in order to accommodate other information and remember it later if subsequent inferencing creates a situation where it can be re-established.

3 Representing knowledge at preliminary stages of engineering tasks

This section describes a framework for representing defaults and preferences within two knowledge structures, models and rules. Models represent information that is available independent of the specific task. Generally, models are based on physical principles, domain theories and contextual parameters. For example, a geometric layout of building elements is a model reflecting physical principles of connectivity; a structural analysis program is based upon models of stress analysis; and a building

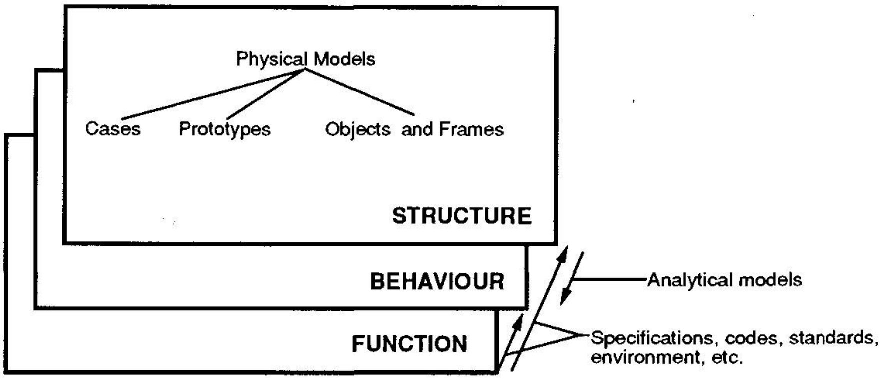

Figure 2: *Types of models along with their planes of abstraction*

code is a model of physical principles and context, such as local requirements and local environmental factors such as wind, earthquake and snow loads.

Models can be expressed on three levels of abstraction - structure, behaviour and function, see Figure 2. Physical models contain information principally on the structural level although important information can exist at all levels [3, 4]. Physical models may contain a great deal of specific information, such as in cases, or relatively little, such as in fundamental hierarchies of elements in a frame system.

Models that contain information that is orthogonal to the structure, behaviour, function planes include analytical models and models of context and user requirements such as building codes, specifications and standards. These models facilitate transformation from one plane to another in order to perform reasoning at the appropriate abstraction. For example, once the structure is defined and loads are known, analytical models are able to transform geometric information in the structural abstraction into stresses and deformations in the behavioural abstraction. Building codes help transform functional requirements into behavioural needs such as maximum deflections and ultimate stresses. These two transformations enable a standard design evaluation to be performed at the behaviour level. In addition to a function-behaviour transformation, codes and specifications include requirements on the geometry of elements and spaces, thereby influencing structural abstractions.

Physical models provide good opportunities for representing general default assumptions. These assumptions can be contained in slots within representations of prototypes and frames. Since cases *already* contain values of parameters, these values act as default values for new contexts where the case is inserted and therefore, default values do not require a special representation.

The second knowledge structure is rules, see Figure 3. Rules contain information that is dependent on engineering decisions made during execution of the task. Rules are divided into two categories, fixed and assumptions. Fixed rules are rules that must be satisfied in order to complete the task. Fixed rules are further divided into two types according to the knowledge they represent - physical principles and technological considerations. Rules related to physical principles represent knowledge about requirements such as geometry - for example, the sum of all bridge spans must equal the length of the bridge. Rules based on technological considerations exist in order to avoid solutions that are technologically infeasible - for example, during construction, helicopter erection is possible only for elements whose weight do not exceed the maximum permitted by the helicopters available to contractors.

Assumptions are the second category of rules. Here lies the knowledge which is most important for effective support of preliminary stages of engineering tasks. As explained in the previous section, two types of assumptions are distinguished - default and preference. Default rules provide default values that are more specific to certain situations than those defaults supplied by physical models. Preference rules introduce knowledge which is difficult to model explicitly. Such knowledge is most easily represented using a rule structure in order to provide the opportunity to identify specific

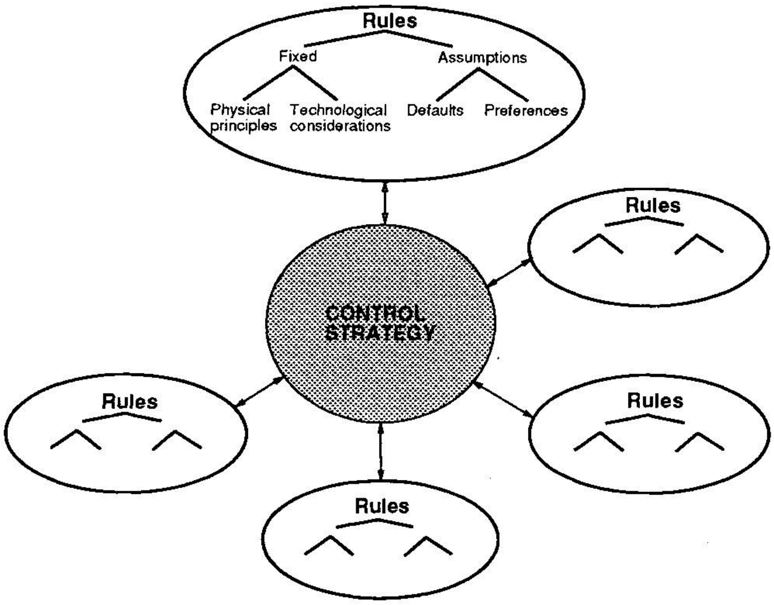

Figure 3: *Rule classifications organized into groups connected to a control strategy*

conditions (in the "if" portion of the rule) for consideration of the preference. An example of the use of assumption rules is given later in the paper.

The ellipses in Figure 3 represent collections of knowledge into groups where all knowledge within an group is consistent. A single domain may have many groups. We believe that such an organization of knowledge is essential to developing a knowledge representation on a large scale. Systems that require creation of one large and consistent knowledge base impose impossible requirements upon developers of the knowledge base and this is only amplified with time, as new information is added and existing information is changed.

The circle labeled control strategy in the center of Figure 3 has two roles. Firstly, it manages conflict resolution when information from different groups is contradictory. The second role determines when each group is added to memory. Since the system resolves conflict through backtracking, relaxing preferences and undoing the effects of default assumptions, the order of appearance of conflicts affects the result of the system. Note that this approach is not a blackboard-type strategy. More details are given in the next section.

4 Reasoning with assumptions

The algorithm employed to control the introduction of rules and the resolution of conflicts is illustrated in Figure 4. Since the focus of this paper is on knowledge representation, the description of this algorithm is brief. More details are given in [5]. Initial conditions related to the task are taken interactively from the user or from a file. These conditions may be used to organize knowledge groups into levels of importance or alternately, the user may select an importance hierarchy from a case base of task plans. These hierarchies are discussed further at the end of this section.

Rules in the lowest level are first added into the rule engine. If the initial conditions do not contain enough information to fire a given rule, default values are taken from the physical model. When information is not explicitly contained in the physical model, analytical models may be employed to derive it. The information inferred by these rules often take the form of constraints which are propagated through one another in order to fix consistent constraints. They are then checked for feasibility since algorithms for propagating constraints on continuous variables may not result in a feasible solution. Considering rules only at the first level, there must be consistent constraints resulting in a feasible solution (by definition, since each knowledge group must contain consistent knowledge) so the feasibility test passes and the next level of rules is loaded into the rule engine.

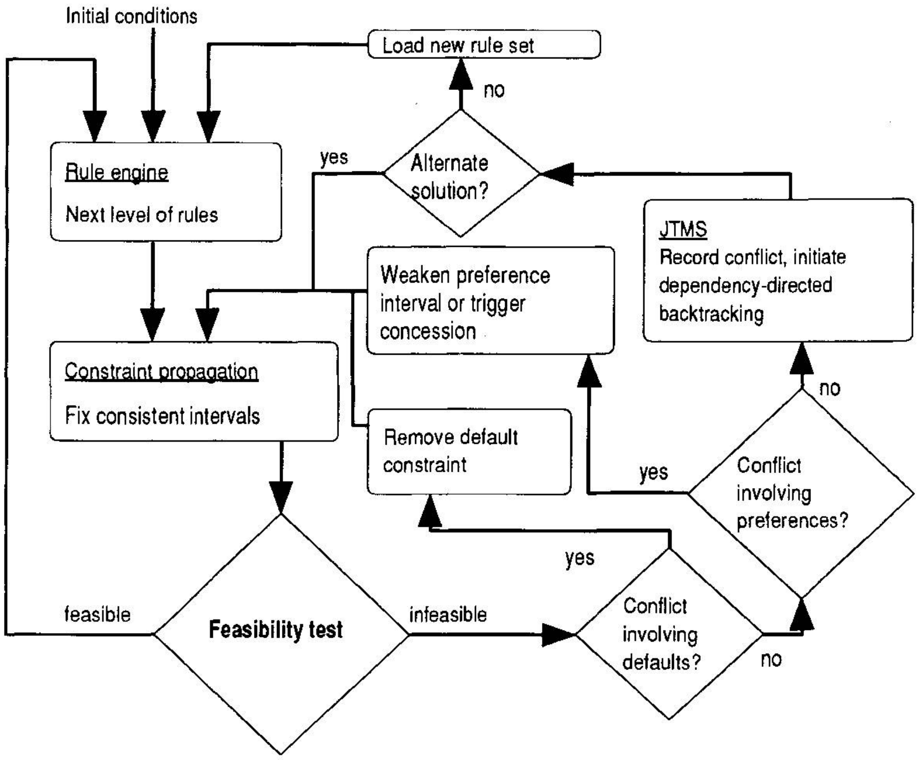

Figure 4: *The algorithm used to reason with defaults and preferences.*

Once the second rule level is processed, infeasible solutions become possible. The system resolves conflicts in the following manner. Justifications for all constraints are recorded. If the conflict involves defaults, the default information is dropped, inferred knowledge is retracted, constraints are propagated again and these are rechecked for consistency. If more than one piece of default information is involved in the conflict, defaults originating from the physical model are dropped first, followed by defaults in the lowest rule level up to the highest.

If a conflict remains after all default information has been dropped, it is examined for the presence of preference rules in the justification list. Here preference rules are weakened in a manner which is not the same as dropping defaults; weakened rules influence valid intervals and they are kept in memory in case a situation arises where they can be fully reinstated. If more than one preference is involved in the conflict, preferences on the lowest levels are weakened first. Since such weakening can result in a drastic change in the solution space, some preferences include concession clauses. In the case of a conflict, concession clauses are used to retract the preference and substitute it with a new rule in order to avoid forcing the system to backtrack to the position where the first preference was inferred. Rules in concession clauses can be of two types, preference or fixed. If preference type concession clauses do not remove the conflict, weakening occurs normally as described above. When the concession clause contains a fixed rule, subsequent conflicts are treated the same way as with fixed rules. This is described next.

When conflicts remain after all preferences have been weakened, the system performs dependency-directed backtracking in order to investigate alternative solution paths. This strategy, along with those that analyse justifications for conflicts, employs a truth maintenance system [6], adapted for intervals of continuous variables. Here the closest available alternate path is taken and often, the system does not require significant backtracking. If an alternate solution path is available, that path is taken and constraint propagation is carried out on the new solution.

If no path is available, a new rule set, consisting of another collection of knowledge groups, is loaded. Rule sets are created for each solution type. For example, different rule sets are needed for different types of bridge designs such as cable stayed, arch, suspension, beam and truss bridges. The

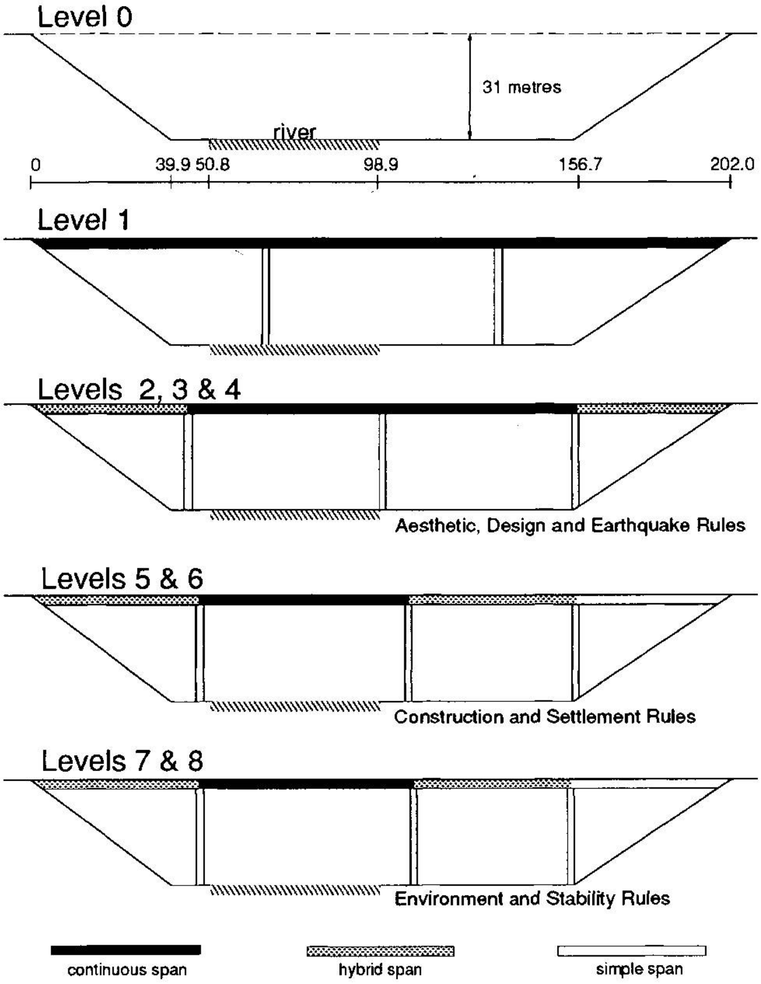

Figure 5: *Evolution of the design of the French Creek Bridge.*

order of priority given to such sets is either predetermined or set through inferences performed on parameters on initial conditions.

Knowledge groups can be organized according to different types of knowledge that influence the task. For example, individual knowledge groups in a bridge design system can refer to design criteria such as aesthetic requirements, construction feasibility, costs, safety, long-term serviceability, detail design and traffic requirements. The order of introduction into the system establishes the relevant importance of each criterion. In this way, the engineer influences how conflict resolution is performed and ultimately, the final solution proposed by the system. Through investigating different hierarchies for each knowledge group, engineers are able to evaluate various solutions which are all consistent with the knowledge contained in the system. Therefore, this knowledge representation, along with the reasoning strategy described in this section, is capable of reflecting the fact that engineers find different solutions for the same problem, depending upon the importance they place on relevant criteria.

5 An example in design

The knowledge architecture described in the previous sections has been implemented in a system called PRELIM, a system for the preliminary design of civil-engineering structures. This example is drawn from a recent bridge project, the French Creek site in British Columbia, Canada. The bridge site is a 202 metre long valley with a 48 metre wide river situated in the second quarter of the valley. The location is remote, access is difficult and the site is situated in a zone of high seismic risk. Furthermore, the construction schedule overlaps with an important reproductive period of the fish population. This

LEVEL	SPAN-1	SPAN-2	SPAN-3	SPAN-4
Level 1 - Start-Up				*note :*
span intervals :	[67.3 67.3]	[67.4 67.4]	[67.3 67.3]	*span values*
span type :	continuous	continuous	continuous	*are in metres*
Level 2 - Aesthetics				
span intervals :	[50.5 50.5]	[50.5 50.5]	[50.5 50.5]	[50.5 50.5]
span type :	continuous	continuous	continuous	continuous
Level 3 - Design				
span intervals :	[48.0 53.0]	[48.0 53.0]	[48.0 53.0]	[48.0 53.0]
span type :	continuous	continuous	continuous	continuous
Level 4 - Earthquake				
span intervals :	[42.6 47.1]	[53.3 58.9]	[53.3 58.9]	[42.6 47.1]
span type :	hybrid	continuous	continuous	hybrid
Level 5 - Construction				
span intervals :	[42.6 47.1]	[53.3 58.9]	[15.0 50.0]	[42.6 47.1]
span type :	hybrid	continuous	continuous	hybrid
Level 6 - Settlement				
span intervals :	[45.8 50.7]	[57.3 63.3]	[45.8 50.0]	[43.0 47.5]
span type :	hybrid	continuous	hybrid	simple
Level 7 - Environment				
span intervals :	[45.8 50.7]	[59.2 63.3]	[45.8 49.1]	[43.0 47.5]
span type :	hybrid	continuous	hybrid	simple
Level 8 - Stability				
span intervals :	[45.8 50.7]	[58.2 63.3]	[45.8 46.8]	[45.3 47.5]
span type :	hybrid	continuous	hybrid	simple
Final Values	48.2	61.7	46.8	45.3
Real Values	47.0	64.5	47.5	43.0

Underlined numbers indicate results after overriding, up-dating, weakening or re-activating a rule

Figure 6: *Modification of span lengths as rule levels are introduced.*

means that no piers or temporary props can be erected in the river and to the right of the river. Finally, the soil conditions are poor with slopes on each side of the valley approaching instability.

The evolution of the design is shown schematically in Figure 5. As the system processes rules from rule level 0 to level 8, parameters such as number of spans, pier position and span type are modified. The evolution of span lengths with rule level is given in Figure 6. A partial list of rules is provided in Figure 7. The physical model for this example is expressed in terms of a frame system. The frame corresponding to this bridge type contains, amongst other information, the following default instances:

- no. of spans : 3
- span type : continuous
- span restrict : 15 to 75 meters

The site configuration is shown at level 0 on Figure 5. Level 0 contains general rules that reflect physical principles such as geometric consistency. Since information about the bridge emerges as the design progresses, PRELIM uses default rules to get started. When level 1 is reached, the "default" or start-up design has 3 continuous spans of equal length. The size and location of the river zone is not yet considered. At level 2, an aesthetics-type rule determines an appropriate number of spans for the height of the valley using aspect ratios and the golden rectangle [7] and resulting in a choice of 4 spans. The rule on level 2 is a preference rule and therefore, the values given by the default rule on level 1 are dropped. Design rules at level 3 assign span factors for each span according to the type of span. Using these factors in a ratio with the total length of the bridge, as well as a rule which allows a certain variation from this factor, preferred intervals for spans are developed, see Figure 6. Since preference rules at level 3 are considered to be more important than preference rules at level 2, the values fixed for spans correspond to the values calculated by the rules at level 3. However, the constraints corresponding to the rule at level 2 are weakened, not dropped. Its effect is reinstated if further inferencing weakens the effect of the rules on Level 3.

CONSIDERATION	TYPE	LEVEL		RULE
General	Fixed	0	If then	number of spans = n sum of the spans (1 to n) = L
General	Fixed	0	If then	span type i is simple and span type i-2 is continuous span type i-1 is hybrid
General	Fixed	0	If then	number of spans = n number of piers = n - 1 and span i+1 = pier i+1 - pier i
Start-Up	Default	1	If then	number of spans = n span i = L/n
Aesthetic	Preference	2	If then	H/L is between 1/15 and 3/5 and H is between 15 and 75 typical span varies from 1.5 to 1.75 H and number of spans = L / typical span
Design	Preference	3	If then	beam type is constant depth and span type i is continuous span factor i is 10
Design	Preference	3	If then	beam type is constant depth and span type is hybrid span factor i is 8
Design	Preference	3	If then	beam type is constant depth and span type is simple span factor i is 7.5
Design	Preference	3	If then	span i has factor i span i is (factor i/ sum of the factors) x L ± 5%
Earthquake	Preference	4	If then	earthquake region is high interior span type is continuous and exterior span type is hybrid
Construction	Preference	5	If then	construction method is launching and beam type is constant depth span i (all spans) < 50
Construction	Preference Concession	5	If then	construction method is launching and beam type is constant depth a maximum of one span can be more than 50 and less than 65
Settlement	Preference	6	If then	soil condition is very poor all span types are simple
Settlement	Fixed Concession	6	If then	soil condition is very poor and river situation in valley is left right exterior span type is simple
Environment	Fixed	7	If then	pier is at edge of sensitive river zone move pier out of zone
Stability	Preference	8	If then	slope is unstable move pier off slope

span : span value in metres   pier : pier position in metres

Figure 7: *A partial list of rules employed.*

At level 4, earthquake requirements change the continuity of the spans - two simple supports at the abutments change the exterior spans to a hybrid span type (simple support at one end and continuous at the other). The system then modifies the span factors for each span in order to reflect these new continuity conditions. This means that the span factors are updated to 8-10-10-8. Construction requirements at level 5 result in the elimination of construction by crane considering soil conditions and access difficulties. Since crane erection is a default instance in the frame, PRELIM now searches for another method. At this point in the development, the system selects the next method of construction from the list of instances in the frame hierarchy. The next construction method is launching. The level 5 rule for launching is a preference limiting the spans to 50 metres. However, a conflict with a physical principles rule at level 0 occurs; the sum of the spans is smaller than the length of the valley. Before weakening what has been inferred thus far, PRELIM looks for a concession clause attached to the most recently added rule which is involved in the conflict. A concession rule is found that accepts one span between 50 and 65 metres. Span 2 is then relaxed to a greater value than 50 and span 3 is limited to 50 after weakening the corresponding rule from level 3.

Settlement considerations in level 6 initially fire a preference for simple supports in order to reduce the stress at the abutments. However, this creates a conflict with earthquake requirements for continuity on level 4. PRELIM finds a concession rule which accepts one simply supported exterior span, the one farthest from the river. Hence span 4 is now simply supported. From a physical principles rule at level 0, span 3 becomes hybrid. Span factors are updated and the design rules from level 3 are re-activated from a weakened state, giving five percent intervals around proportions of 8-10-8-7.5. At this point, construction criteria are still respected but some earthquake criteria are weakened. This demonstrates the persistence of preference rules. Rather than having their effect eliminated entirely, as is the case for default rules, they remain in memory and are re-established when justifications for weakening disappear.

At levels 7 and 8, PRELIM loads rules from groups that focus upon pier position. At level 7, an environmental concern for spawning beds prevents a pier positioning in the river and 11 metres to the right of the river. Although no direct conflicts occur, a level 0 rule relating spans as a function of pier positions up-dates and narrows the range of possible values for spans 2 and 3. At level 8, there is a strong preference to avoid placing a pier on an unstable slope. As was the case for level 7, the ranges of possible values, this time for spans 3 and 4, are modified.

Final values of the spans are selected by applying the design rules similar to those on level 3 (without the five percent tolerance rule), and are adjusted with fixed rules of physical principles. The values obtained compare well with the values adopted for the the project in reality. Therefore, using PRELIM, a rough simplification of the actual preliminary design process results in values that are close to those for the as-built bridge.

6 Conclusions

Engineers often make important decisions when information is not reliable. Therefore, assumptions are an essential part of knowledge employed during preliminary stages of engineering tasks. When this knowledge is divided into defaults and preferences and then integrated into models and rules organized into knowledge groups, there is much potential for representing assumptions in a flexible manner. With such a representation, iterative reasoning strategies use priorities attached to knowledge groups to resolve conflicts rationally and simulate the evolution of decisions as more information becomes available. Preliminary engineering tasks can thus be supported in situations where knowledge is incomplete and contradictory.

Acknowledgements

Funding for this research was provided by the Swiss National Science Foundation. The authors would like to thank Boi Faltings for useful discussions and Djamila Haroud for development and implementation of the reasoning algorithm. Also, the contribution of Dr. Peter Buckland, Buckland and Taylor Consulting Engineers, North Vancouver with respect to the knowledge used for the design case described in this paper is gratefully recognized.

References

[1] **WOOD, K.L and ANTONSSON, E.K.** "Computation with imprecise parameters in engineering design: background and theory" J. of Mechanisms, Transmission and Automation in Design, 111, 1989.

[2] **GARRETT, J.H. et al** "Engineering applications of neural networks" Journal of Intelligent Manufacturing, 1992.

[3] **GERO, J.** "Design prototypes: A knowledge representation schema for design", AI Magazine, Vol 11, No 4, 1990.

[4] **UMEDA, Y. et al** "Function, behaviour and structure" Applications of Artificial Intelligence in Engineering V, Springer-Verlag, 1990.

[5] **FALTINGS, B., HAROUD, D. and SMITH, I.** "Dynamic constraint propagation with continuous variables" 10th European Conference on AI, ECAI'92, Vienna, 1992.

[6] **DOYLE, J.** "A truth maintenance system" Artificial Intelligence, 12, 1979.

[7] **LEONHART, F.** Bridges, aesthetics and design, Verlags-Anstalt, Stuttgart, 1982.