Structural Monitoring using Engineer-Computer Interaction

Ruth Stalker[1] and Ian Smith[2]
[1] Computing Department, Lancaster University, LA1 4YR, UK
[2] Institute of Structural Engineering and Mechanics, EPFL-Swiss Federal Institute of Technology, Lausanne, Switzerland

---

[1] Dr. Ruth Stalker, Computing Department, Faculty of Applied Sciences, Engineering Building, Lancaster University, Lancaster, UK, LA1 4YR
ruth@comp.lancs.ac.uk, Tel: +44 (0) 1524 593116, Fax: +44 (0) 1524 593608

To be published in Artificial Intelligence for Engineering Design Analysis and Manufacturing, Special Issue, September 2002

Abstract

Engineer-Computer Interaction (ECI), is a new sub-domain of Human-Computer Interaction specifically tailored to engineers needs. ECI i) uses an information classification schema, ii) provides a modular approach to task decomposition, and iii) integrates standard engineer characteristics and working procedures into software. A software toolkit that interprets monitoring data taken from bridges was developed according to ECI guidelines. This toolkit was given to engineers for testing and evaluation. Empirical evaluation using questionnaires was performed. Results show that this ECI software corresponds to engineers needs and that the ECI approach has the potential to be applied to other engineering tasks.

To be published in Artificial Intelligence for Engineering Design Analysis and Manufacturing, Special Issue, September 2002

1. Introduction

The first application of information technology (IT) to structural engineering was an analysis program for plane frames (proposed in 1956 at Manchester University, U.K.). This became the starting point for much research into the use of computers for structural engineering as it illustrated the utility of IT for analysing large structures (Grierson, 1996). Although in 2000, computers are ubiquitous in structural engineering, engineers remain frustrated with the inadequacy of computer support. Fifteen commercially available software statistical packages were reviewed in Burr (1999) with the conclusion that none were suitable for engineers. This was because they did not offer the functions that engineers desire to perform certain tasks.

Engineers must perform tasks using incomplete knowledge, problem specific characteristics and context dependency (Salvaneschi, 1996). Thus, tasks are difficult to model completely, although there have been attempts (Fenves, 1989). Instead of modelling everything (including engineering expertise), a more practical approach is to enable engineers to interact with the computer to add or delete information as desired in order to make software calculations more compatible with reality. Engineers are legally responsible for their decisions and therefore, they need software which provides realistic solutions and in which they have confidence (Smith, 1996). Thus, the interaction between computer and user becomes just as important as the algorithms (Wegner, 1997). Even though Human-Computer Interaction (HCI) is a growing domain and its importance has been recognised in structural engineering, it remains a secondary consideration (Anumba, 1994). Reciprocally, HCI does not address engineers as a specific group of users with their own particular needs.

In this article, Engineer-Computer Interaction (ECI), and its application to the domain of structural monitoring, is described. ECI, is a sub-domain of HCI tailored particularly to the needs of structural engineers. ECI supports the design and development of software for engineers as described in Section 2. A description of the testing and evaluation of ECI, using software developed for structural monitoring and diagnosis, is given in Section 3. Section 4 contains the results of evaluating the software by questionnaire which show that engineers are provided with more appropriate decision-support. Finally, conclusions and future work are given in Section 5.

2. Engineer-Computer Interaction

Engineer-Computer Interaction (ECI) is defined in Stalker (2000) as:
a sub-domain of human-computer interaction for the design, evaluation and implementation of interactive decision-support systems for engineering tasks.

ECI is composed of the following three aspects:

- Organisational Schema: A function, behaviour and structure schema of engineering information which represents important stages in a structure's life cycle.

- Task decomposition: ECI task decomposition (TD) identifies sub-tasks which have been specifically chosen to incorporate iteration, multiple solutions, comparison and viewpoints into the information transformations. These transformations occur during the tasks that are identified in the organisational schema.

- Engineer Identikit: Task decomposition is supported by a generic representation of engineers. This representation enables easy assembly of a graphical user interface (GUI) through implementation of appropriate features.

2.1 Organisational Schema

In Gero (1990) a schema was proposed in order to consider a designed artefact in terms of function, which is the semantics of a design; behaviour; and structure - the syntax of a design. Here the schema is augmented in order to represent temporal aspects. The subscripts $_t$, $_0$, and $_{t^0}$ indicate time before the artefact physically exists. In this paper bridges are used as artefact examples. Thus, the subscripts

$_t$, $_0$, and $_{t^0}$ indicate time before the bridge physically exists. This augmented schema is shown in grey in Figure 1. Tasks such as design, analysis, formulation, synthesis and construction transform information from one category to another. In ECI the schema is extended in order to represent the whole lifecycle of a bridge and tasks such as monitoring, model correction, intervention and dismantling are added. These tasks use the subscript $_{t^n}$ to indicates the many iterations the bridge design may have gone through before it is built (extensions are shown in black in Figure 1).

Function F_0 is a set which refers to structural requirements and reflects objectives such as strength, serviceability, security and durability of structures. Such objectives cannot be directly transformed into a set representing structural description possibilities S_{t^0} without first anticipating desired, or expected behaviour. Therefore, functional objectives are formulated in terms of expected behaviour B_0. The task of synthesis uses expected behaviour B_0 to provide a set of structural descriptions S_{t^0}. Thus, many structural descriptions S_{t^0} may be formulated and iteratively refined. The transformation, or synthesis, of expected behaviour to a structural description S_{t^0} is a difficult task. Structural description S_{t^0} is a geometrical description of the artifact with the topological configuration of types of elements, such as a beam or trusses. S_{t^0} contains material properties and environmental effects, such as loading.

The iterative process of F_0 to B_0 to S_{t^0} to B_{t^0} and comparison of behaviours B_0, B_{t^0} is performed through the tasks of formulation, synthesis, analysis, and traditional evaluations until a suitable structural design description is decided upon. The task of construction uses the selected structural design description S_{t^0} in order to create an actual, physical, structure S_{t^n}.

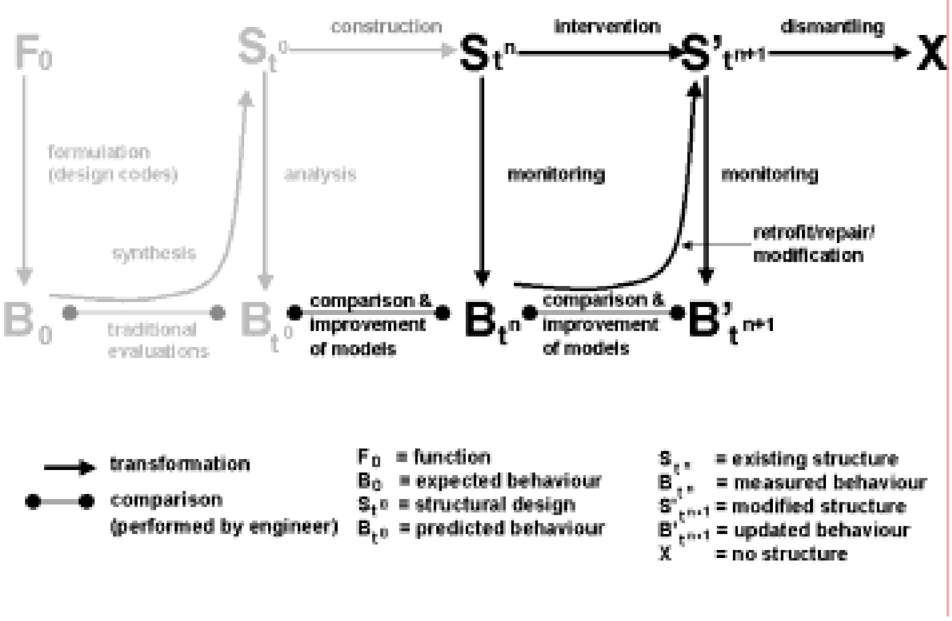

Figure 1: Engineering information is classified into categories in terms of function, behaviour, and structure. Engineering tasks, such as synthesis, analysis and construction transform this information from one category to another. Monitoring, modification and prediction are added to the schema. Transformations are iterative (not shown in figure)

The monitoring transformation maps the physical structure S_{t^n} to measured behaviour B_{t^n}. Comparison between measured and predicted behaviours (B_{t^0} and B_{t^n} respectively), should lead to improved structural representations of both the physical structure $S'_{t^{n+1}}$ and the analytical representations B_{t^0}. Updated behaviour $B'_{t^{n+1}}$ and modified structure $S'_{t^{n+1}}$ are later stages in the structural life-cycle. X illustrates the end of the structure's life.

The use of the schema enables a classification of information and task to be more simply translated into a software structure to be implemented in a computer. Each task is an iterative procedure which must be decomposed into manageable sub-tasks for ease of use and to fully exploit available information.

2.2 Task Decomposition

Each task transformation (as defined in Section 2.1) is divided into five modules (Figure 2). These modules have been specifically chosen in order to encourage the employment of multiple solutions, comparison and viewpoints. These aspects have been used in engineering tasks such as design and analysis (Stalker, 2000).

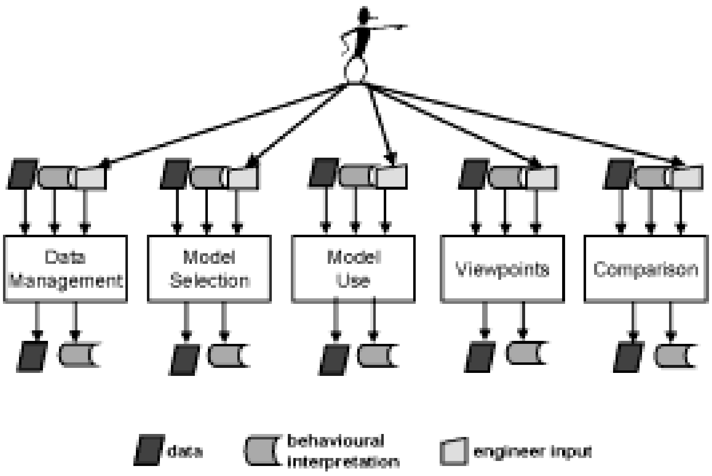

Figure 2: Task decomposition is represented by 5 modules. Each module can receive input from the engineer and they do not have to be used chronologically

2.2.1 Data Management

The purpose of this module is to examine the validity of software input. It is important that the information input (in whatever form: database, matrices, or topology declarations) is realistic.
The data management module (DM) enables the semi-automation of validating this input in the following ways:

- Heuristically. A heuristic approach is computationally represented through constraints by the specification of boundaries. Input values must lie within these boundaries in order for them to be acceptable.

- Statistically. This is a more historical approach where the mean and standard deviations of input sets are calculated using previous input sets. A normal distribution of errors is assumed. Using simple statistics, confidence levels are set in order to judge the validity of input set. Examples of such statistical approaches can be found in Stroud (1995).

- Visually. Input is visualised using an annotated visualisation. Data points are labelled with pertinent information. Any doubtful values are highlighted by the system. This enables the engineer to interactively accept and reject values just by observing the input. Such a presentation with highlighted aberrant input is a form of active-decision support (Smith, 1996).

- Plug-ins. Software has been developed for the validation of computer-aided design (CAD) input and finite-element analysis input (CADFIX, 2000). Therefore, one method would be to reuse these software packages.

The engineer decides how to use the input:

- Admission: the engineer accepts the input as is, including aberrant values.
- Omission: the engineer deletes aberrant values or asks the data management module to do so automatically. This leads to an incomplete input but there are still enough data points for the set to be useful.
- Rejection: the engineer is notified that the quality of the data-set is poor. This normally means that there are too many aberrant values or the set does not have enough values to be of use. This may be either the original input or the resulting input after having passed through the omission stage.

During model formulation and selection, decisions related to model sensitivity may affect data management methods.

2.2.2 Model Selection

A space of solutions is more useful than just one solution. Therefore, the model selection module (MS) contains a choice of models which represent various combinations of behavioural assumptions for the engineering task to be performed.

This approach uses model based reasoning (MBR) which uses models abstracted from reality, formulated by engineers or taken from engineering literature, as opposed to heuristics given by experts (rules). Where possible, engineering models are founded on sound physical principles. In this way various representations of structure lead to the calculation of different structural behaviours which are then employed by engineers for subsequent decision making. During model formulation and selection, decisions related to model sensitivity affect data management methods. The approach of “when" to use a certain model is not addressed because the assumption of the model selection model is that engineers know which models they want to use.

Assumption	Fundamental	Accuracy	Verified	Hypothesised
Pure beam bending	X		X	
Continuous over supports	X		X	
Linear-elastic behaviour	X		X	
Twin column support		X		
No relaxation in prestressing force	X			X
Rigid beam-column connection		X		X
Cracks at the supports	X		X	
Rotational springs within span	X		X	
Deep beam hypothesis		X		
Load carrying capacity of deck		X		
Pin roller at supports		X		X
Point loads	X			
Constant temperature gradient	X			
Linear temperature variation				X

Table 1: Modelling assumptions for the Lutrive bridge (after Raphael, 2000)

Table 1 is a list of possible modelling assumptions for the Lutrive bridge, Switzerland. These assumptions are organised into the following four categories:

1. Fundamental: indicates whether a given assumption is fundamental to the creation of a minimal model.
2. Accuracy : indicates whether a given assumption will affect the accuracy of the model.
3. Verified: indicates whether a given assumption is made after verifying/analysing structural behaviour.
4. Hypothesised: indicates whether a given assumption is assumed due to a lack of knowledge.

These assumptions were used for structural monitoring purposes in Raphael (2000) and were originally taken from design models and used for analysis (Robert-Nicoud, 2000).
It can be seen by looking at the crosses in each box how models are constructed. For example, the assumption of pure beam bending is fundamental to the creation of a minimal model and can be verified by the engineer.

These models are represented in a GUI by engineering plans and symbols.

2.2.3 Model Use

In order to support current practice, the model-use module makes parameters accessible so that engineers can modify them. This is because the behaviour of a structure, which is represented by a model, may change significantly when model parameters are changed. Table 2 contains model parameters that can be changed. The same assumption classification is used as before. Hence, model parameters are categorised according to whether they: are fundamental to model creation; affect model accuracy; can be verified; and are hypothesised.
As model parameters are available for engineers to modify, MBR becomes a semi-automatic and explicit approach and thus it is more accessible to engineers than heuristic rules.

Assumption	Fundamental	Accuracy	Verified	Hypothesised
Varying moment of inertia		X		
Column stiffness during bending		X		
Plan curvature of beam		X		
Young's modulus				X
Hinges at mid-spans		X		X
Non-linear material		X		
Geometric non-linearity		X		
Support settlement	X	X		
Weight/position of point loads	X			
Distributed wheel loads			X	
End conditions		X	X	X
Stiffness coefficients				X

Table 2: Model parameters are explicitly represented so that engineers may change these parameters and experiment with different behaviours (after Raphael, 2000)

2.2.4 Viewpoints

Viewpoints is a module which gives the engineer the possibility to select partial models or sub-sets of data in order to consider models and data from alternative views. This module enables the exploration and exploitation of both data and model-use. This module depends on the following factors:

- Focus: Through viewpoint fixation task objectives become the focus of the task. For example, task objectives can be the realisation of elegance, efficiency, economy, and utility (Billington, 1995; Shea, 1997). Without focus objectives may be overlooked.

- Exploration: Through the consideration of the space of solutions from a specific point of view, the space is altered. This can be by selecting part of a structure or part of a data set. This is viewpoint exploration. Alternatively, a new space is created for investigation purposes (Navinchandra, 1991). This may be performed through the addition of deletion of data or by looking at particular structural aspects.

- Exploitation: Through viewpoint exploitation, the space of solutions is examined in order to make the best possible use of the information available (Smithers, 1998).

2.2.5 Comparison

Comparison, the last module in the task decomposition, is simplest way to evaluate the validity of a solution. It is a form of passive-decision support (French, 1986). Comparisons may be performed between tasks or between iterations of one task. In the case of task comparison, results from another task can be read in and a comparison performed. An engineer may want to compare predicted behaviour to measured behaviour (B_{t^0} and B_{t^n}) in order to measure the disparity between the two behaviours so that models of predicated behaviour can be improved and used more accurately in the future. Comparison between iterations of one task may give multiple solutions which have been produced for example, by changing parameters in the models. These multiple interpretations are considered in order to exploit the information available. Each time a new interpretation is created, the comparison module stores this interpretation with all relevant information, such as the input and the model with its parameters. These are kept until an engineer deletes them.

The validity of a solution is judged using the same methods as the DM module for judging the validity of system input. These methods are: heuristic, statistic “best-fit", visual, and plug-ins. Other forms of comparison may be useful (e.g. non-dimensional). However, these are the methods are left to engineer choice. In order to access the TD approach it is necessary to have an easy-to-use GUI in the form of the engineer identikit.

2.3 Engineer Identikit

The engineer-identikit is the part of ECI concerned with GUI development specific to engineers needs. It takes a user-centred design based approach (Preece, 1997; Dix, 1998) in order to produce a GUI that is comfortable and intuitive for an engineer to use.

Engineers are trained to perform tasks and meet specific objectives. They often use formalised procedures for confronting problems. They use explicit knowledge, which takes a hierarchical form, in order to classify information during engineering tasks. Engineers make trade-offs between competing objectives such as time, cost, and quality. During discussions they describe physical behaviour or properties of artefacts using mathematical formulae. They often employ a specialised graphical language which uses predefined combinations of symbols and graphical representations thereby communicating without ambiguity. Finally, engineers usually have specific tasks to perform and want to do them in different ways. This engineer identikit (Figure 3) should be looked upon as GUI toolbox builder where engineer aspects are used in software systems as desired. Part or all of it may be used.

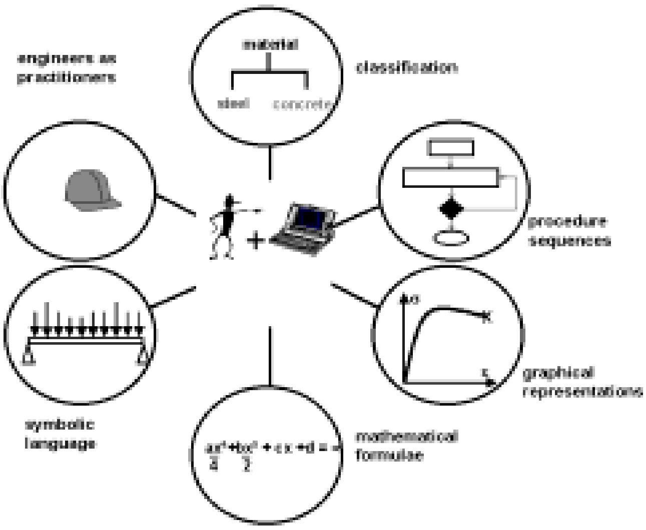

Figure 3: Engineers use formulae and graphical representations to work with knowledge. These characteristics are reproduced in ECI software

2.3.1 Computer Literacy

Engineers are generally computer literate. By 1975 computing was on the syllabus for engineers at universities and books that used computing examples such as McCormack (1975) were recommended reading. Thus, manual calculation for engineering tasks is no longer a feasible alternative. The ECI approach is grounded in the belief that good solutions are the combined effort of engineer experience and computer support.

2.3.2 Classification

Structural engineering uses a more rigorous terminology and information classification than domains such as computer science. Classification may be based on material properties, which in turn determines material behaviour, acceptable loads on materials, and the tests which should be performed. This information is imperative because it dictates how engineers should use materials. It is also useful for talking to other engineers and is, therefore, a practical way to represent information in a computer.

Figure 4 contains a hierarchical classification of a bridge. This is a typical decomposition strategy for representing bridges during design and such decompositions should be incorporated into software systems (Boulanger 1997).

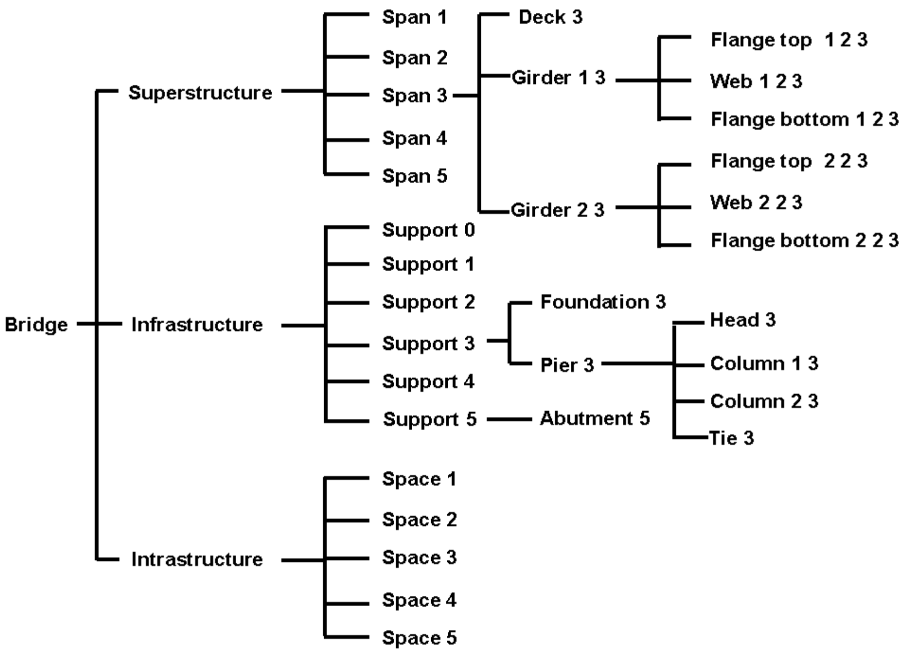

Figure 4: A hierarchical classification of a bridge, after Boulanger, (1997)

2.3.3 Task Objectives

Engineers are trained to perform a finite number of tasks and to realise objectives in a certain manner. Figure 5 contains two examples of computer generated roof trusses. Both trusses are functionally sound but neither satisfy engineers' objectives which are dictated by society and culture. The left hand side truss is much heavier than the one on the right hand side which is unsymmetrical and has more cross sections and joints. From an aesthetical viewpoint the left hand side is more pleasing.

Figure 4: Two computer generated trusses illustrate task objectives of elegance and utility, after Shea, (1997)

Billington (1995), classified task objectives into efficiency, economy and elegance. Shea (1997) added utility. These objectives are weighted differently in each project and may even be dictated by politics (Boulanger, 1994). Thus, it is important that engineers have the opportunity to interact with software in order to choose task focus and to prioritise task objectives. In this way, it is the engineers who steer the calculation process to achieve desired results instead of the computer providing the engineer with results which they will not use.

2.3.4 Procedures

Engineers often follow predefined procedures when performing tasks. For example, the tensile test (Megson, 1987). These procedures are found in the codes of practice which gives guidelines for tasks related to structures throughout their life-cycles. These rules are codes and procedures for design, evaluation, and testing in order to ensure structures meet with criteria such as safety and serviceability. Procedures are rigorously structured. Such an approach inspired software engineers to be equally rigorous in designing software (Pressman, 1994).

2.3.5 Graphical Representations

Graphical representations are used in scientific fields such as physics, mathematics, and engineering. An image may contain much information and in structural engineering certain graphical representations are familiar to engineers. They often sketch the bending moment or deflection of a beam using a 2D graph. Figure 5 contains the graphical representation of the direct relationship between stress and strain for ductile material. Therefore, it is advantageous to include such representations in the engineer identikit. Engineers identify the meaning behind these graphical representations and can see if calculations are giving the desired results.

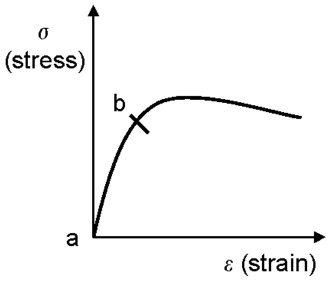

Figure 5: Graphical representations are useful and recognisable, after Megson, (1987)

2.3.6 Mathematical Formulae

When in discussion, engineers describe the physical behaviour and properties of artifacts through mathematical formulae. For example, the deflection of a simply supported symmetrical beam of length l which bends under a uniform load q whilst its plane cross-sections remain plane and normal to its longitudinal fibres can be described mathematically as a quartic relationship (Equation 1)

$$\frac{qx}{24EI}\left(x^3 - 2lx^2 + l^3\right) \quad (1)$$

where x is the distance from the left-hand support. Most other HCI user groups prefer mathematical formulae and algorithmic details to be hidden and results to be represented graphically, whereas engineers enjoy manipulating formulae symbolically and quantitatively.

2.3.8 Symbols

Alongside formulae and graphical representations are the combinations of predefined symbols engineers employ in order to attempt to communicate without ambiguity. These symbols are precise. Figure 4 illustrates two similar structures with differing supports. The structure on the left has two triangles, each triangle represents a support and one support has two circles. The structure on the right does not have any circles and triangles. The omission of circles and triangles indicates that the supports are fixed. Not only are fixed supports more expensive than hinged supports, they change the behaviour of the proposed structure. Thus, circle and triangle removal will produce a different structure. Such an error is semantic and not merely syntactic. Symbols are precise. The incorporation of such symbols into graphical user interfaces builds on existing engineering work procedures and provides interface transparency.

Figure 4: The omission of triangles and circles changes structural behaviour

2.3.9 Engineers as Practitioners

Engineers are practitioners. Their characteristics are similar to other practitioners such as, in business, law and medicine. Society trusts them to perform in a responsible manner. Although laws may fix limits on their activities, they have much freedom to identify creative solutions. Engineers a) read plans, b) have a legal responsibility, c) make trade-offs, and d) need flexibility.

2.3.9.1 Plans

Due to the size and complexity of structures, many plans (geometric descriptions on paper) are needed to provide accurate geometrical representations. For example, during construction of a mid-size bridge, approximately 20 kilograms of plans are produced. These plans should be used in graphical user interfaces.

2.3.9.2 Legal Responsibility

Civil engineers have been responsible for their work since Egyptian times. Legal responsibility is widely acknowledged by engineers. With such responsibility it is important that any tools which engineers use, such as computers, must support them in an understandable way and make provision for engineers to modify calculations should the need arise. Engineers are often required to justify their decisions during technical litigation processes. This is not the case for many other groups of users.

2.3.9.3 Trade-offs

Engineers make trade-offs between competing objectives such as time, cost and quality. These objectives are dictated by the economic and social constraints within which the engineer must work. Different design objectives of elegance, economy, efficiency and utility produce very different designs. Also, not all domains are influenced so pointedly by additional factors such as the availability of resources and political issues (Boulanger, 1994). Such factors cannot be easily modelled in a computer. Engineering experience can translate factors into a value a computer can work with. Therefore, the engineer must be given the opportunity to interact with software in order to choose task focuses and to prioritise task objectives. Good interaction between the engineer and the computer allows engineers to steer the calculation process in order to achieve the most reasonable results within given constraints.

2.3.9.4 Flexibility

Finally, engineers have specific tasks to perform and the order in which steps of a task are executed may change from engineer to engineer. Therefore, flexibility of approach is often an important requirement within and between tasks. This flexibility should be reflected in software (Boulanger, 1997). Software should not impose a specific sequence of steps if it is to be adopted by engineers.

2.4 Application of ECI to engineering tasks

The application of ECI to a given task is approached in 3 steps as shown in Figure 5.

1. The relevant part of the organisational schema is selected in order to have a foundation for the software.
2. Modules are chosen and used as necessary depending on the task. Not all modules may be needed.
3. Desired engineering characteristics are chosen from the engineer identikit and added into the software in order to support the modules.

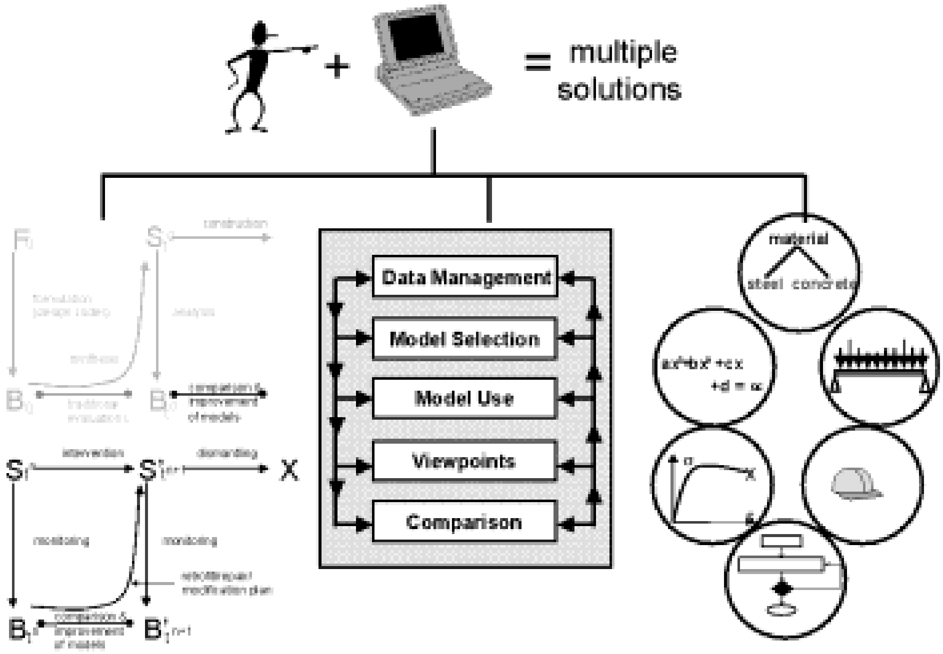

Figure 5: ECI contains an organisational schema, task decomposition and an engineer identikit

ECI is representative of a decade long trend to replace automated systems with a collaboration between the engineer and the computer. In this way, engineer experience and the computer's computational capacity are used together. Interactive systems are more useful than automated reasoning since they provide transparency and enable collaboration between engineer and computer. The following section illustrates the application of the ECI blueprint to the task of structural monitoring.

3. Structural Monitoring Tool Kit

3.1 SMTK Organisational Schema

Structural monitoring and the interpretation of data is the transformation link between the existing structure (S_{t^n}) and measured behaviour (B_{t^n}) as shown in Figure 1 and described in Section 2. Monitoring data interpretation takes measured results and endeavours to find an explanation for structural behaviour.

The use of the schema makes the difference between expected behaviour (B_{t^0}) and measured behaviour (B_{t^n}) explicit. The assumptions made during the task of analysis, in particular loading assumptions, are used to calculate expected behaviour (B_{t^0}) which may be very different to measured behaviour (B_{t^n}). These two factors are the main motivators for structural monitoring. The use of the organisational schema makes it easier to define software objectives.

The schema also enables the engineer to explicitly identify the types of information SMTK needs to interpret monitoring data successfully. This information is as follows:

- Monitoring data: this data may be read in either directly on-line whilst on-site, or off-line from a database. IT represents raw structural behaviour and therefore, it is the data which is to be interpreted. This information is treated by the data management module and represented in the organisational schema as (B_{t^n}).

- Structural knowledge: this knowledge describes the structure which is being monitored in terms of its dimensions, or co-ordinates of length, breadth and height. Plans of the structure which sketch these co-ordinates are scanned in at the same time so that the engineer has a visual representation of the structure. The structure's co-ordinates are used for calculation in the model-use and comparison modules. The plans and the co-ordinates are used together in the viewpoints module for visual representation and calculation. This knowledge makes up part of existing structure (S_{t^n}). More information for S_{t^n} is expressed as loading and environmental considerations.

- Monitoring equipment: this information describes the position of monitoring apparatus on or in the structure. The equipment description is used for calculation in the model-use module and for visualisation purposes in the viewpoints modules.

- Interpreted information: Previous interpretations are read into SMTK.

3.2 Task Decomposition for SMTK

The ECI task decomposition of five modules was instantiated as SMTK modules. Each module is described in the following paragraphs.

3.2.1 Data Management

This module enables the validation of the current data-set. A given data-set is one group of measurements. If a bridge is equipped with 30 deformation measuring instruments and each instrument provides a value at a given time, the result would be 1 data-set of 30 points. If the same action was performed 8 times in 1 day then 8 data-sets would be available. Each data-set can be viewed as a snapshot of structural behaviour at a specific time. Data quality is important, as some measurements may be false due to measuring apparatus problems such as a mechanical fault or incorrect employment. Values can be judged in two ways: heuristically and statistically. The data management module heuristically judges the aberrant values (outliers). The history of each value is considered as shown in Figure 6. The box in the figure illustrates the constraints placed on acceptable values for this data point. Points outside of the box are brought to the engineer's attention by SMTK because they have been judged as invalid.

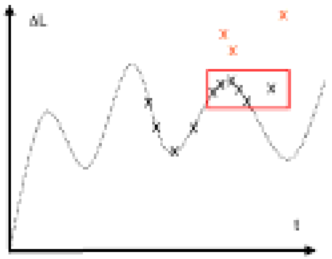

Figure 6: Judging data

This box changes considering the history of the data point. If the data point is within a series which increase or decrease in value and the data point in question does not, SMTK highlights the point so that the engineer is aware of its possible aberrant value.

3.2.2 Model Selection

Each bending deformation measurement can be treated as a data point which is plotted against a curve representing the expected deformation (Vurpillot, 1999). In other words the data is reduced using a

model. These curves, however, are only approximations. Different assumptions lead to very different deflection curves. Thus, it is better to have a selection of deflection curves from which to choose.
The models contained in this model selection module should be chosen, or created, by the engineers who are going to use SMTK with respect to the structure which they are monitoring. The models are represented in the module by symbols and mathematical formulae from the ECI identikit.

3.2.3 Model Use

The model-use module enables the engineer to explore a chosen model in detail. The choice of which parameters the system makes explicit should be taken at the same time that the model-selection module is formulated. Possible parameters which may have an effect on the behavioural interpretations are stiffness coefficients (e.g., the moment of inertia), and types of loading.

3.2.4 Viewpoints

This module uses a plan of the structure and monitoring apparatus. The engineer may click on the structure and the monitoring apparatus in order to select areas of interest. For example, there may be several types of monitoring apparatus in one structure. The engineer may select just one type of monitoring data from one piece of apparatus in order to interpret these results alone. Alternatively, the engineer may want to look at specific parts of the structure and consider its behaviour independent of global structural behaviour.

3.2.5 Comparison

Once calculations have been performed they may be stored for comparison with other interpretations of a particular data-set. Comparison may be performed according to different criteria:

- Time for comparison on snapshots of behaviour at certain times. This is a common approach with engineers who may want to compare a structure's behaviour on a monthly or even yearly basis.
- Multiple models. Different models give different behavioural interpretations. Therefore, an engineer may want to try out several models in order to find a behaviour which fits the data.
- Multiple apparatus. Each apparatus may produce different data. The engineer may want to compare measurement results.

Once several functions for structural behaviour are found the comparison module allows the engineer to compare these functions statistically: SMTK uses statistics to find which function fits the data best, or visually: The engineer may be able to see which function is more representative of structural behaviour.

Each curve represents:

- A function for curvature which is calculated in order to calculate the function for deformation.
- A function for deformation which describes the global deformation of the structure which is being monitored.
- A data-set which is reduced in order to calculate the above functions.
- An indication of which model the deformation function represents.
- The model parameters contained in the deformation function which the engineer specified during the data interpretation.

Other interpretations from data-sets may be read into this module using this format so that comparison may take place. Many comparisons are possible, e.g. non-dimensional comparison. Thus, it is for the engineer using the system to decide what types of comparison are appropriate.

3.3 SMTK Engineer Characteristics

The SMTK-GUI was made up using the following characteristics from the engineer identikit as follows:

- **Task objectives:** The task objectives of structural monitoring are to find an interpretation of structural behaviour.

- **Formal procedures:** The models in the following example (Section 3.4) are based Bernouilli beam theory (simple, or pure, beam bending).

- **Graphical representations:** A scatter plot illustrates a data-set and a graph represents the calculated functions for global deflection. These are presented to the engineer in the data-manager module and the model-use module.

- **Mathematical formulae:** The functions which represent global deflections are described mathematically alongside the graphical representation in the model-use module. Behavioural assumptions which are made by different models are mathematically described in the model-selection module.

- **Plans:** The viewpoints window contains a plan view and a cross-sectional view of the structure to be analysed. Engineers use these plans to make correlations between the monitoring data and the structure.

- **Symbols:** The model-selection module uses symbols to represent structural behaviour. This enables the engineer to understand the behavioural assumptions contained in the models.

3.4 SMTK for the Versoix Bridge

SMTK was used to interpret monitoring data which was taken from the Versoix bridge which supports part of the dual carriageway (N1) close to Geneva airport, Switzerland. It was built in the 1960s and is made up of two parallel bridges. Each bridge is made of two prestressed beams which support a reinforced concrete slab of 30cm. In the interests of safety the road was enlarged on both sides to create a “hard shoulder". Concrete was added on both sides of both bridges in direct contact with the old concrete as shown on the right side of the figure. There were concerns about the interface between old and new concrete, concrete shrinkage and the spatial deflection of the bridge due to these extensions. Thus, during the extension process the bridge was equipped with a network of SOFO fibre optic sensors in order to measure local deformations. A description of how the SOFO sensors were developed and how they work can be found in Inaudi (1997). Two beams of the Versoix bridge were equipped with 96 fibre optics. The first beam (A) was equipped with 5 cells of sensors. The second beam (B) was equipped with 7 cells of sensors. Each cell contains 8 sensors as shown by the cross-section schema in the viewpoints window of SMTK in Figure 8 (at the end of Section 3.4.6). The data is read by the monitoring equipment and stored in a database called SOFO-DB (Inaudi, 1997). SMTK accesses the database and treats each data-set as a separate file. SMTK was implemented in OpenGL and C/C++ and runs on a Silicon Graphics Indigo 2 (SGI). The software is presented module by module and ECI engineer identikit characteristics are highlighted under each module heading.

3.4.1 Data Management

A data-set is one set of measurements taken at a specific moment in time on the bridge. Therefore, a given data-set contains a maximum of 96 data points. If many readings were taken in one day the data-set is referred to by its date and time. This enables comparisons to be made at various times of day using the comparison module. The viewpoints window (Figure 8) illustrates the layout of each cell of sensors. In order to calculate the radius of curvature successfully the cell needs to produce a minimum of 2 data points. These values must have 1 from the top and 1 from the bottom. In a normal data-set the sensors at the top of the beam give positive values and the sensors at the bottom give negative value. The data-set is plotted on a graph in the module. The y axis measures positive and negative deformations. The x-axis is the length of the beams. Each line represents one data point. It is annotated with the name of the sensor and its corresponding local deformation. The data points are plotted on the x-axis to represent their position in the beams and on the y-axis to represent their deformation. The data management window in Figure 8 illustrates a typical Versoix bridge data-set. Visualising the data in this way makes it easier i) to see aberrant values; ii) allows the engineer to become familiar with a scattering of data and iii) more easily identify bad data-sets. The engineer may delete data as required. Clicking on a data point displays the information in the bottom right hand corner of the module. However, if an engineer clicks on the delete button and then clicks on a data point, the data point is removed from the data-set. This removal is reflected in the data manager window, in the viewpoints window, and in the end calculation.

It has been shown that a removal of a total cell of data values may result in a 10% difference in the accuracy of the result (Vurpillot, 1999) and this becomes more obvious when visualising data in such a way.

3.4.2 Model Selection

The model-selection module contains three models. One model was specifically developed for the interpretation of data from the Versoix bridge (Vurpillot, 1999) based on pure beam bending. The second model extends this base model to deal with point loads over a specific sensor point. The third model emulates shear.

3.4.2.1 Pure Beam Bending

This model assumes that the law of Bernoulli is satisfied. That is to say, the plane cross-sections of the beam remain plane during bending. The model requires that each beam is divided up into sections (as shown in Stage 1 of the left-hand side of Figure 7). It is assumed that each section has a moment of constant inertia, a uniform load across its length and supports only at its ends. Each section is made up of a group of cells. These cells contain a minimum of two sensors which are placed at parallel to an assumed neutral axis and are used to calculate the mean radius of curvature (Stage 3, Figure 7):

$$\frac{1}{r_m} = \frac{\Delta L_2 - \Delta L_1}{(y_1 - y_2)L} \quad (2)$$

These local curvatures are used to describe the curvature of the whole section. Therefore, the local radii of curvatures are fitted to a curvature function using a polynomial of the appropriate degree:

$$P_2(x) = ax^2 + bx + c \quad (3)$$

The polynomial has three unknowns. Therefore, only three independent measurements (i.e. cells) are necessary to express the curvature function of a single beam section of the Versoix. If there are more than three cell results, which is the case for the Versoix, the system of equations should be solved by least-squares. A summary of this model is given in Figure 7. For a more detailed description see Vurpillot (1999).

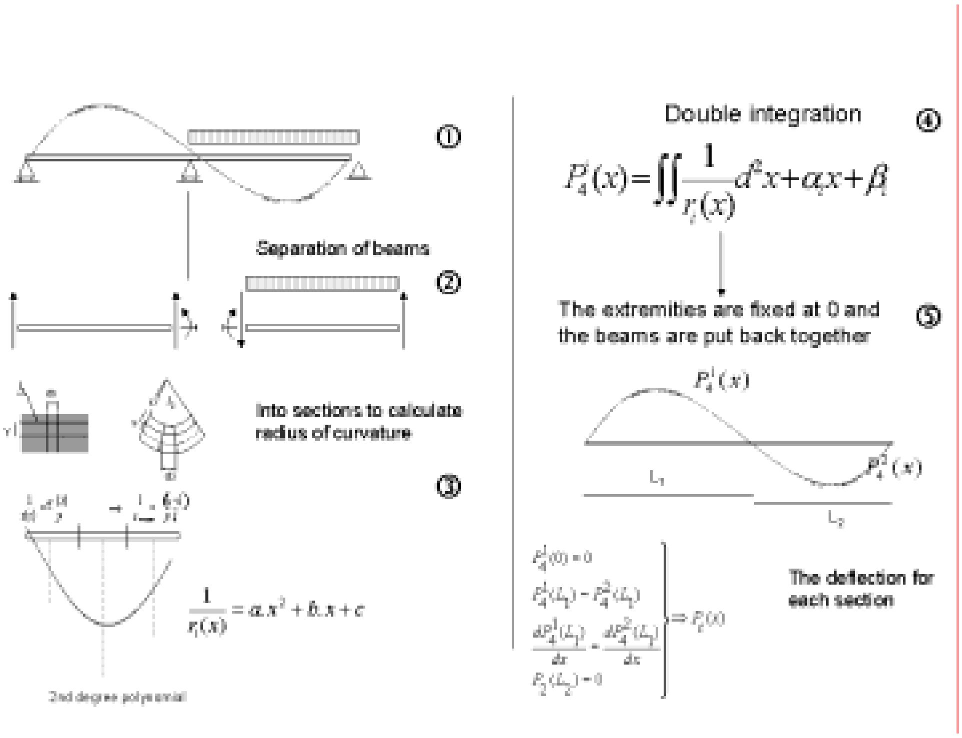

Figure 7: A summary of the data reduction model used in SMTK for the Versoix bridge after Vurpillot, (1999)

3.4.2.2 Point Loads

Local change in curvature deflections due to point loads may be underestimated using the previous model for transforming deformations into beam deflections. Therefore, a correction for additional deflection of point loaded beam as proposed in (Timoshenko, 1970) is added to the model. The conditions for this model correction validity are that the length of the SOFO sensor is four times greater than half the depth of the beam and that the point load is over the mid-point of a SOFO sensor.
In order to account for the increase in curvature caused by this point load compared with curvature caused by distributed loading, an extra term is added when calculating the radius of curvature and Equation 2 becomes:

$$\frac{1}{r_m} = \frac{\Delta L_2 - \Delta L_1}{(y_1 - y_2)L} + \alpha \frac{P}{EC^2} \quad (4)$$

in which α is a numerical factor varying along the beam, value is given in (Timoshenko, 1970), *P* is the force per unit thickness of the beam (e.g. wheel load), *E* is Young's modulus, and *C* is half the depth of the beam. This is useful for modelling point loads (e.g. a lorry during test conditions is parked over a SOFO sensor). An illustration of the test conditions of such a test that was performed in Switzerland is given in Perregaux (1998).

3.4.2.3 Shear

The shear model emulates structural behaviour using the same approach. The assumptions are that there is uniform loading on the beam and the SOFO sensors are placed parallel to the neutral axis.
By adding a term which is referred to as the effect of shearing force (Timoshenko, 1970) Equation 3 is rewritten as:

$$\frac{1}{r_m} = \frac{\Delta L_2 - \Delta L_1}{(y_1 - y_2)L} + \alpha \frac{q}{EI} C^2 \left(\frac{4}{5} + \frac{r}{2} \right) \quad (5)$$

where q is the uniform load per unit length, E is Young's modulus, I is the moment of inertia, and α is Poisson's ratio.

Supports

Following the first group of assumptions as described by each of the models above: *Pure Beam Bending; Point Loads; Shear*, the second group of assumptions are the support conditions. These are chosen by the engineer who clicks on the desired model and the support assumptions and this activates the data interpretation process.

3.4.3 Model Use

The model-use module is used to refine the following model parameters in order to change the interpretation. These parameters have already been used in the equations in the previous section. These variables are made accessible to the engineer through the SMTK-GUI.

Degree of Polynomial: The degree of polynomial may be specified by the engineer.

Position and weight of point load: The position and the weight of the point load (P) can be specified in order to indicate where the results of the SOFO sensor should be adjusted.

Moment of Inertia: The moment of inertia (I) can be input by the engineer and is used for calculation purposes for the shear model.

Boundary conditions: By default it is assumed that the displacement boundary conditions are zero However, an engineer may want to declare them to be less than zero in order to represent support settlement.

The model-use module has a graph. The x-axis shows the length of the beams and the y-axis illustrates the global deflection. The deflection function has a mathematical label.

3.4.4 Viewpoints

The viewpoints window is used by the engineer to select areas of interest in the bridge. Structural images were scanned from the original plans in order to keep within current working practice of an engineer. The engineer may click on parts of the plans as they are representative of the structural knowledge contained in the system. For example, areas of the structure can be selected so that the monitoring data from this area alone is considered. An engineer may want to concentrate on the supports or in the middle of the beam where one expects displacement to be quite small.

The viewpoints window in Figure 8 illustrates the visual presentation of the Versoix bridge in the viewpoints module. At the top of the window there is a cross-sectional view which illustrates the position of the eight sensors. The sensors, numbered 1 to 8, were placed in this manner in order to measure the behaviour of the new concrete (1 and 2), the interface between the old and new concretes (3 and 5, 4 and 6) and to obtain an overview of global structural behaviour (7 and 8). In Figure 8 the plan view contains highlighted cells which have data measurements. The cell which is highlighted with a long line (cell A1) corresponds to the cross section view above.

In this way solutions can be taken, explored and investigated. This can be done by:

- Reducing or increasing the number of data points
- Selecting one or two beams
- Selecting various cells instead of using all of them

The SOFO sensors which are active and provide values for the current data-set are highlighted in the bridge plan.

3.4.5 Comparison

Each curve represents a data-set, a function of displacement, end conditions, moment of inertia etc. Hence, an engineer clicking on a curve will receive information about the curve. SMTK indicates the “best-fit" to the engineer. The “best-fit" is calculated by summing the square of the errors in order to find the smallest error margin.

3.4.6 Book Keeping

A sixth window referred to as bookkeeping is added to the five windows which represent the five task decomposition modules. It keeps a record of all the actions taken during the use of the toolkit. Book-keeping is a simple window which displays a list of all actions taken by the engineer since software start-up. In this way the engineer has a guide to the calculations SMTK has performed so far and if data has been added or deleted.

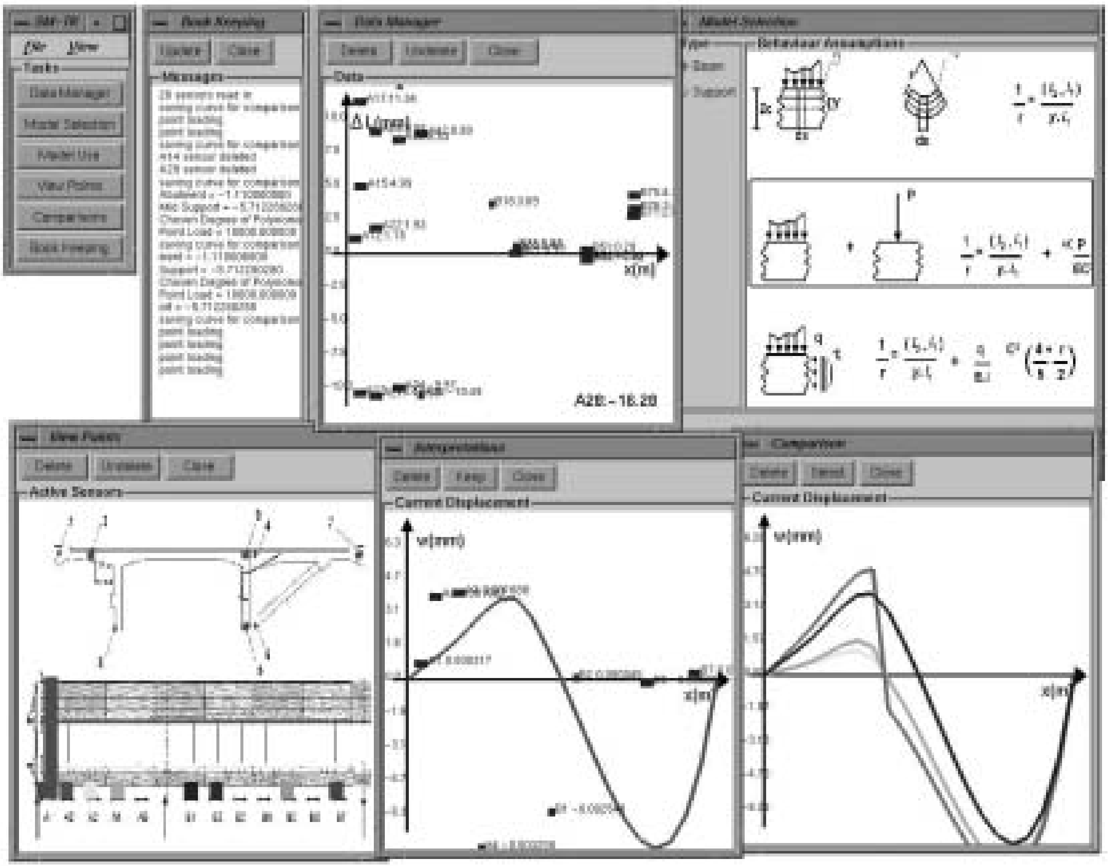

Figure 8: A screen shot of the SMTK in use. In the top left hand corner of the figure is the SMTK toolbox through which the engineer opens and closes windows. The windows are from top left clockwise book keeping, data manager, model-selection, comparison, model use, and viewpoints

4. Results

The QUIS™ (QUIS) questionnaire was used to assess subjective user satisfaction with SMTK. The results of a pilot study were inconclusive (see Stalker, 2000). Thus, QUIS was used in a SERV-QUAL manner (Zeithaml, 1997) so that user expectations of what SMTK should do could be plotted directly against user perceptions of what SMTK actually does in order to derive conclusions about user satisfaction with SMTK. Two groups of eight people were questioned. One group contained structural engineers. The second group was made up of “non-engineers” from various disciplines.

4.1 A SERVQUAL Analysis of Engineers and Non-Engineers using SMTK for Monitoring Data Interpretation

The questionnaire was divided into 6 parts. This article presents only parts 2 and 6 of the results. The rest of the results analyse in more detail each part of the ECI framework. More results and discussion can be found in Stalker (2000). Parts 2 and 6 contained the same questions with only the verb "expect" in part 2 replaced by "perceived" in part 6. Table 3 contains both questions numbers (e.g. 2/6.1) and verbs (e.g. expect/perceive) and both ends of the measurement scale (e.g. bad/excellent). Thus, each question can be derived from the header in Table 3 and a question number and entry. For example, looking at the table it is possible to derive that Question 2.1 read "Do you expect the software to be frustrating/excellent?" and Question 6.1 is "Do you find the software frustrating /excellent?. At this point the user had a scale of 1 to 5 (frustrating-1 to excellent-5) and had to tick a number. Part 2 was given to both groups before using SMTK in order to measure expectations. Part 6 was given after use of SMTK in order to measure software perceptions. Figure 9 shows the mean of each question response. E-Expect represents engineer expectations before SMTK use. N-Expect represents the non-engineer group's expectations of SMTK. E-SMTK represents engineer perceptions of the software. NE-SMTK represents non-engineer perceptions. The question numbers in Figure 3 correspond to the question numbers in Table 3. Table 3 contains a summary of the questions the users had to answer on a scale of 1 to 5 before (expectations) and after (perceptions) using SMTK.

Question No	**Overall expectations/perceptions of software for monitoring**
2.1/6.1	Expect/Perceive system to be bad/excellent
2.2/6.2	Expect/Perceive system to be frustrating/satisfying
2.3/6.3	Expect/Perceive system to be dull/stimulating
2.4/6.4	Expect/Perceive system to be difficult/easy
2.5/6.5	Expect/Perceive system to be not-useful/useful
2.6/6.6	Expect/Perceive system to be rigid/flexible
2.7/6.7	Expect/Perceive system to be non-pertinent/pertinent
2.8/6.8	Expect/Perceive system to be not user-friendly/user-friendly
2.9/6.9	Expect/Perceive system to be non-reliable/reliable

Table 3: Parts 2 and 6 of the QUIS questionnaire applied in a SERV-QUAL manner to measure monitoring data interpretation software SMTK

The midpoint of the rating scale (3) was used as the criterion. Therefore, if the question response was above 3 it was perceived to be better than average. In general (Figure 9) it can be seen that engineers expectations and perceptions of SMTK were both above average. In contrast, the non-engineers perceptions were below average. The results show that engineers were more demanding in their software expectations than their non-engineer counterparts. The line which represents their expectations is higher than the non-engineers expectations. However, engineers did not expect to find the software to be the most stimulating, flexible, and user-friendly (Questions 2.3, 2.6, 2.8). Therefore, their expectations of the capabilities of proposed software would be in some way based on their experiences with current software and it may have been difficult to imagine software to be flexible and stimulating. Nevertheless, they expected new software to be excellent, satisfying, easy and useful.

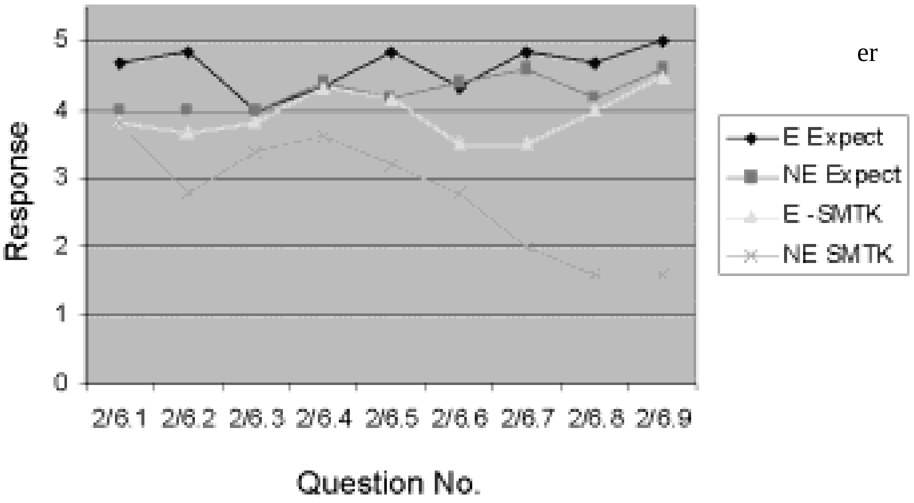

Figure 9: A SERVQUAL means analysis of engineer and non-engineer group expectations and perceptions of SMTK for monitoring data interpretation

The non-engineers were not quite as enthusiastic in their expectations of software that would be useful for interpreting monitoring data. Although, the task had been explained to them and the majority of the “non-engineer" group had experience of interpreting data in a similar way (i.e., looking for trends in the data and curve fitting) they had difficulty anticipating software that could offer sufficient support for such a task. Thus, answers to Questions 2.1, 2.2, 6.1 and 6.2 showed that non-engineers were not expecting SMTK to be excellent or stimulating. In terms of stimulation (Questions 2.3 and 6.3), engineers were not as satisfied as expected. This was because some of the engineers found the models in SMTK to be unsatisfactory. However, their critique of the models in SMTK illustrates that the model assumptions in the system were clearly represented so that engineers could make comments on the model choice at such an early stage in software evaluation. Also, engineers did not find SMTK to be as flexible and as pertinent as they wanted. For example, SMTK does not link up to a finite elements package and therefore, it is unable to provide analytical and experimental comparison. This was not the goal of SMTK. Nevertheless, the observation was made that a link between structural analysis and monitoring could easily be established – comparison that is rarely performed. SMTK led engineers to imagine ways in which it could be more useful and extended to perform many tasks that they currently have difficulty performing. Finally, engineers expected to be stimulated by SMTK and expected SMTK to be easy to use which they found to be the case (Questions 2.3 and 6.3 and 2.4 and 6.4).

Non-engineers found SMTK to be relatively (in comparison to the engineers) frustrating (Questions 2.2 and 6.2), non-pertinent (Question 2.7 and 6.7), non-user friendly (Question 2.8 and 6.8) and non-reliable (Question 2.7 and 6.7). SMTK uses symbols which non-engineers did not understand. It has graphical representations, symbols, and plans of structures that are not the tools non-engineers use daily. Thus, these are ECI attributes are difficult for non-engineers to appreciate. SMTK was described as non-reliable because non-engineers did not know what sorts of results to anticipate. Therefore, it was difficult for them to judge whether SMTK gave “right" (relevant) answers.

Five of the non-engineer question means were below the average point (3). These results illustrate that SMTK is not suitable for non-engineers.

Engineers reacted more favourably to SMTK than non-engineers because the software corresponded to their needs through the employment of ECI. Engineers were able to understand and use the engineering terminology in the system. Their appreciation of the separation of structural and behavioural information and the explicit rendering of model assumptions was illustrated by the positive response. They found that it was easier to perform monitoring data interpretation using SMTK. Moreover, they had many suggestions to improve this instantiation of ECI for SMTK application. These suggestions were: a) the linking of SMTK to a finite elements package so that more sophisticated models could be used; b) a link so that comparison between analysis and monitoring could take place; and c) how ECI could be applied to other engineering tasks. (These ideas as are developed in Chapter 7 of Stalker, 2000.)

Their main criticism of SMTK was the choice of models used in the system. However, the transparency of the interface, which was constructed using the ECI identikit, is illustrated with this criticism because it was possible for the engineers to understand the models almost immediately.

Non-engineers liked SMTK less than engineers because they had difficulty understanding symbols and found graphical representations less useful. However, they did appreciate that the task of interpreting monitoring data was well structured by the use of ECI task decomposition, which lent an impression of simplicity to SMTK.

5. Conclusions

ECI represents a contribution to the need for a specific approach to the design, implementation, and evaluation of interactive decision-support systems for engineering tasks. This paper has presented the application of ECI to the task of structural monitoring. A toolkit called SMTK (Structural Monitoring Tool Kit), that interprets bridge monitoring data, was developed according to ECI. SMTK was given to engineers to evaluate and their reactions, collected by questionnaire, were analysed. Results show that this software is closer to engineers' expectations of good software than the packages they currently use. SMTK offers appropriate decision-support that enables engineers to perform more tasks in a more satisfactory manner than is currently possible. Future work will involve the application of ECI to the tasks shown in the ECI organisational schema.

Acknowledgements

The authors would like to thank David Brown, Simon Bailey, Esther Gelle, Chimay Anumba, and Paul Xirouchakis for their comments and feedback on this research. This work was performed whilst Ruth Stalker was at the Institute of Structural Engineering and Mechanics, EPFL. Logitech SA and Silicon Graphics Inc. are also thanked for supporting this work.

References

Anumba, C. J. (1994). “Considerations in User-Interface Design For Knowledge-Based Systems", Artificial Intelligence and Object Oriented Approaches for Structural Engineering, pp 47-51.

Billington, D. P. (1995). “The Tower and the Bridge: The New Art of Structural Engineering", Basic Books, New York,

Boulanger, S. (1997). “Preliminary Bridge Design Navigation Tool for Novices", Ph.D. Thesis 1735, Swiss Federal Institute of Technology,

Boulanger, S. & Smith, I. (1994). “Knowledge Representation for Design Tasks", Knowledge-Based Systems, 7, pp 161-168,

Burr, A, Craig, B. A., Chung, C., So, T., Deveau, T., Shirer, D. (1999). “Programs for Statistics and Data Analysis" Computing in Science and Engineering January-February, pp 16-21.

CADFIX. (2000) http://www.fegs.co.uk/cae.html

Dix, A., Finlay, J., Abowd, G., Beale, R. (1998) “Human-Computer Interaction”, Second Edition, Prentice Hall Europe.

Fenves, S. J. (1989). “Expert Systems: Expectations versus Realities", IABSE, Bergamo, pp 1-19

French, S. (1986). “Decision Theory: An Introduction to the Mathematics of Rationality", Ellis Horwood, Chichester.

Gero, J. (1990). “Design Prototypes: A Knowledge Representation Schema for Design", AI Magazine, 11,4, pp 26-48.

Grierson, E.E. (1996). “Information Technology in Civil and Structural Engineering Design: Taking Stock to 1996", CIVIL-COMP Ltd, Edinburgh, Information Processing in Civil and Structural Engineering Design Civil-Comp Press, pp. 7-15.

Gaylord, C.N. (1990). “Structural Engineering Handbook”, 3rd Edition, McGraw Hill.

Inaudi, D. (1997). “Fiber Optic Sensor Network for the Monitoring of Civil Engineering Structures", PhD Thesis No 1612, Swiss Federal Institute of Technology.

McCormack, J., “Structural Analysis”, 3rd edition, Intext Education, New York.
Megson, T. H. G. (1987). “Strength of materials for civil engineers", 2nd edition, Edward Arnold, Newcastle-upon-Tyne.

Navinchandra, (1991). “Exploration and Innovation in Design: Towards a Computational Model", Symbolic Computation, Artificial Intelligence, Springer.

Perregaux, N. (1998). “Pont de la Lutrive -N9. Equipement et Analyse du Comportement au Moyen du Systeme de Mesure a Fibres Optiques SOFO", Diploma, Swiss Federal Institute of Technology.

Pressman, R, (1994). “Software Engineering: A Practitioner’s Approach”, Third Edition, European Edition, McGraw-Hill.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T. (1997) “Human-Computer Interaction”, Addison-Wesley.

Raphael, B., Smith, I. (1998). “Finding the right model for bridge diagnosis", Artificial Intelligence in Structural Engineering, Computer Science, Heidelberg, pp 308-19

Robert-Nicoud, Y., Raphael, B., & Smith, I. (2000). "Decision support through multiple models and probabilistic search", Proceedings of Construction Information Technology 2000, Icelandic Building Research Institute, Reykjavik, 2000, pp 765-779

Salvaneschi, P., Cadei, M., & Lazzari, M. (1996). “Applying AI to Structural Safety Monitoring and Evaluation", AI in Civil and Structural Engineering IEEE Expert, Intelligent Systems and their Applications pp 24-34

Shea, K. (1997). “Essays of Discrete Structures: Purposeful Design of Grammatical Structures by Directed Stochastic Search", Ph.D. Thesis, Carnegie-Mellon University.

Slaughter, L., Norman, K., & Shneiderman, B. (1995). “Assessing Users' Subjective Satisfaction with the Information System for Youth Services (ISYS)", CAR-TR-768, Human-Computer Interaction Laboratory, Department of Psychology and Computer Science, University of Maryland, United States.

Smithers., T. (1998). “Towards a Knowledge Level Theory of Design Process", Artificial Intelligence in Design, Kluwer Academic Publishers, pp 3-21.

Smith, I. F.C., (1996). “Interactive Design: Time to bite the bullet", Information Processing in Civil and Structural Engineering Design, Civil Comp Press, Edinburgh, pp 23-30.

Stalker, R.A., (2000). “Engineer-Computer Interaction for Structural Monitoring”, PhD Thesis No. 2199, Swiss Federal Institute of Technology.

Stalker, R., & Smith, I. (1998). “Structural Monitoring: Decision Support through Multiple Data Interpretations" Artificial Intelligence in Structural Engineering, Computer Science, LNAI 1454, Springer Verlag, Heidelberg 1998, pp 474-478.

Stroud, K. A . (1995) “Engineering Mathematics”, Fourth Edition, MacMillian Press Ltd, England.

Timoshenko, S. P. , & Goodier, J. N. (1970). “Theory of Elasticity", 3rd edition, McGraw-Hill.

Vurpillot, S. (1999). “Analyse Automatisee des Systemes de Mesure de Deformation pour l'Auscultation des Structures’’ PhD Thesis No 1982, Swiss Federal Institute of Technology.

Wegner, P. (1997) “Why interaction is more powerful than algorithms", Communications of the ACM, (40), 5, 80-91

Zeithaml, V. A., Parasuraman, A., & Berry, L. L. (1990) “Delivering Quality Service: Balancing Customer Perceptions and Expectations” , New York: The Free Press.