AN INTERACTIVE TOOLKIT FOR STRUCTURAL MONITORING

Ruth Stalker and Ian Smith

Institute of Structural Engineering and Mechanics (ISS-IMAC),
EPFL-Federal Institute of Technology,
CH-1015 Lausanne, Switzerland

{ruth.stalker, ian.smith}@epfl.ch

Abstract: This paper presents a perspective of Compatible Human-Computer Interaction (CHCI) for engineers. This perspective includes their needs, and helps tailor software according to engineering tasks. We refer to this as Engineer-Computer Interaction (ECI). The ECI framework is instantiated as a software toolkit for structural monitoring and an evaluation of work so far is presented.

1. Introduction

Artificial intelligence methods are used in engineering to support complex tasks [1,2]. Through applications, it has been observed that computers cannot model all information due to problem specific characteristics and context dependency. Hence, user interaction is essential [3]. The domain of human-computer interaction (HCI) focuses on user aspects of computer systems [4]. Developing HCI methods is just as important as developing underlying algorithms for usable software [5]. However, proposals from HCI are not appropriate for engineers. Commercial packages do not fulfil their needs; even statistical software is lacking in adequate support for simple tasks such as non-linear regression [6]. Engineers are legally responsible for the decisions that they make therefore, they will not use software which does not provide clear and open support.

Although software compatibility is a major concern due to different platforms and distributed resources, compatibility between the human and the computer is rarely considered. The “Islands of Automation” metaphor illustrates “islands” of software and proposes “boats” of standardised exchanges of information [7]. Unfortunately, the user is not present in the diagram. Ethnography is the scientific description of races and people, their customs and habits. While this approach has been adopted in HCI in order to design systems for users, it is inadequate for defining system specifications [8]. A more effective approach involves obtaining actual experience in the relevant field.

Engineers are users with special needs and should be treated as such. In Section 2 we present a framework which replaces the word, human in HCI with the word, engineer (Engineer-Computer Interaction) in order to encourage compatibility between the engineer and the computer. This framework is instantiated as a software toolkit for structural monitoring as described in Section 3 and evaluated in Section 4, with a brief discussion on future work.

2. A framework for Engineer-Computer Interaction

2.1. Characteristics of an Engineer

Engineers are computer literate and aware of the advantages of using a computer. They are trained to perform tasks in order to realise objectives. They use formalised procedures for confronting problems and 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 and they often employ specialised graphical languages that use combinations of predefined symbols in order to communicate without ambiguity. There are many types of graphical languages in engineering; some examples are provided in Figure 1. Finally, engineers usually have specific tasks to perform and want to do them in different ways.

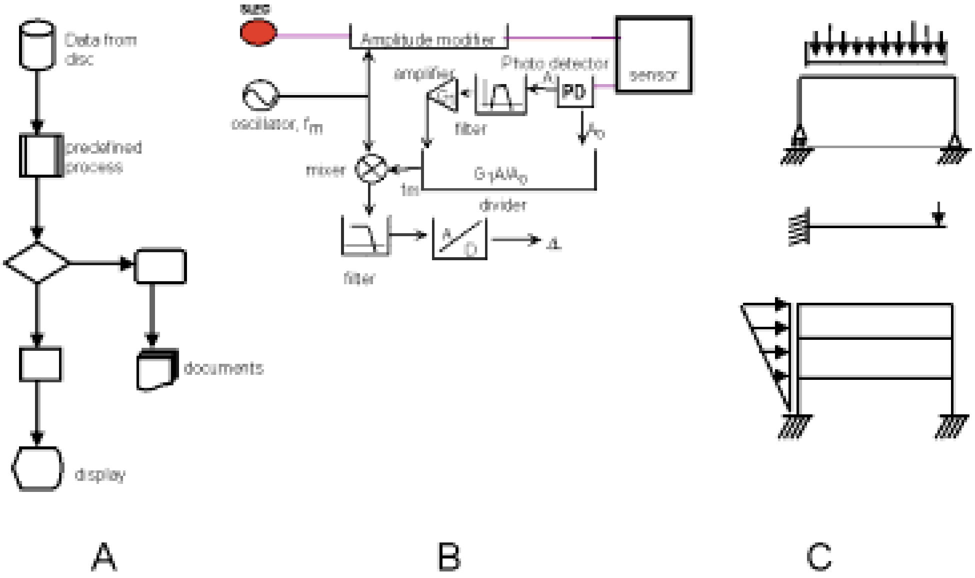

Figure 1: Graphical languages: A) Software engineering: data is interpreted from a disc through different processes and output to a screen. B) Electrical engineering: the layout of an apparatus, which interprets a light signal from a sensor. C) Examples of structural engineering. Each language is unique and symbols are not interchangeable from one language to another

2.2. Engineering Information and Tasks

Engineering systems involve incomplete data, use explicitly defined models and integrate various software technologies [9]. Engineering information can be divided into categories of function, behaviour, and structure. Tasks, such as design, analysis, and measurement transform information from one category to another. The function and purpose of an artefact, in this article, have the same objectives. Example objectives are strength, serviceability, security and durability. The predicted behaviour calculated by analysis and the measured behaviour is obtained through monitoring. These behaviours are considered together for the comparison with expected behaviour, which is often obtained from design codes. In the absence of measurements, predicted behaviour is often too conservative. This is due to compounding of assumptions which ultimately leads to the excessive cost of structures [10]. Structure includes geometry, topology and types of elements, such as a beam or trusses. This may also include a description and layout of the monitoring apparatus. This definition of function, behaviour, and structure is an enhanced version of the schema proposed by [11] and is shown in Figure 2.

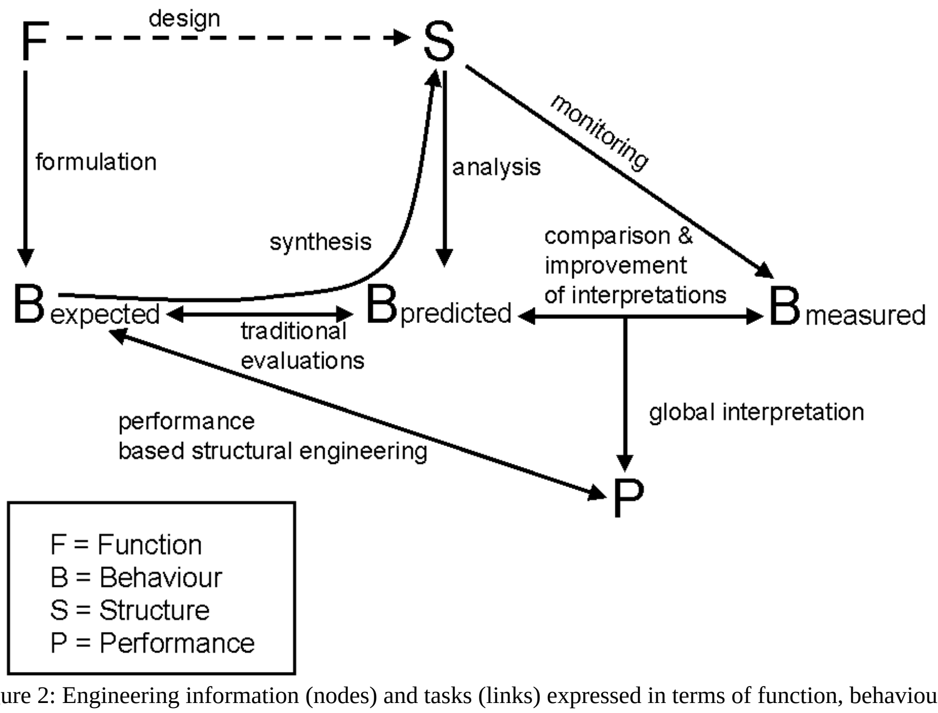

Figure 2: Engineering information (nodes) and tasks (links) expressed in terms of function, behaviour, and structure, extended [11]. This representation is used to support engineering tasks such as design, analysis and monitoring

2.3. Framework

The goal of Engineer-Computer Interaction (ECI) involves the determination of appropriate support so that engineers are helped and not hindered. Implementation of ECI requires a framework for the design, evaluation, and implementation of decision support systems for engineering tasks. The ECI framework contains the enhanced function, behaviour, structure definitions (Figure 2) along with the special characteristics of engineers as described in Section 2.1. This ECI framework is shown in Figure 3.

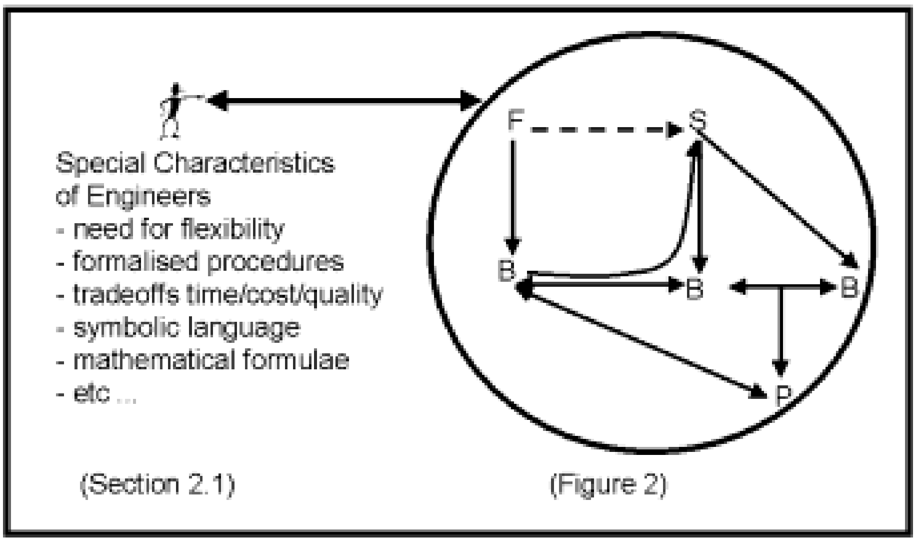

Figure 3: The Engineer-Computer Interaction framework is made up of engineering tasks that link information expressed as function, behaviour and structure coupled with the special characteristics of engineers

3. An ECI toolkit for Structural Monitoring

This toolkit has been developed in C/C++ with OpenGL and X/Motif and on Silicon Graphics for ease of portability. It is currently being tested with SOFO, a fibre optic monitoring system which measures local deformations [12]. The toolkit has five independent software modules which are called methods. Input is the opening of one or more files. Output is presentation on the screen, referred to as presenting and the creation of files for inclusion in reports is known as reporting.

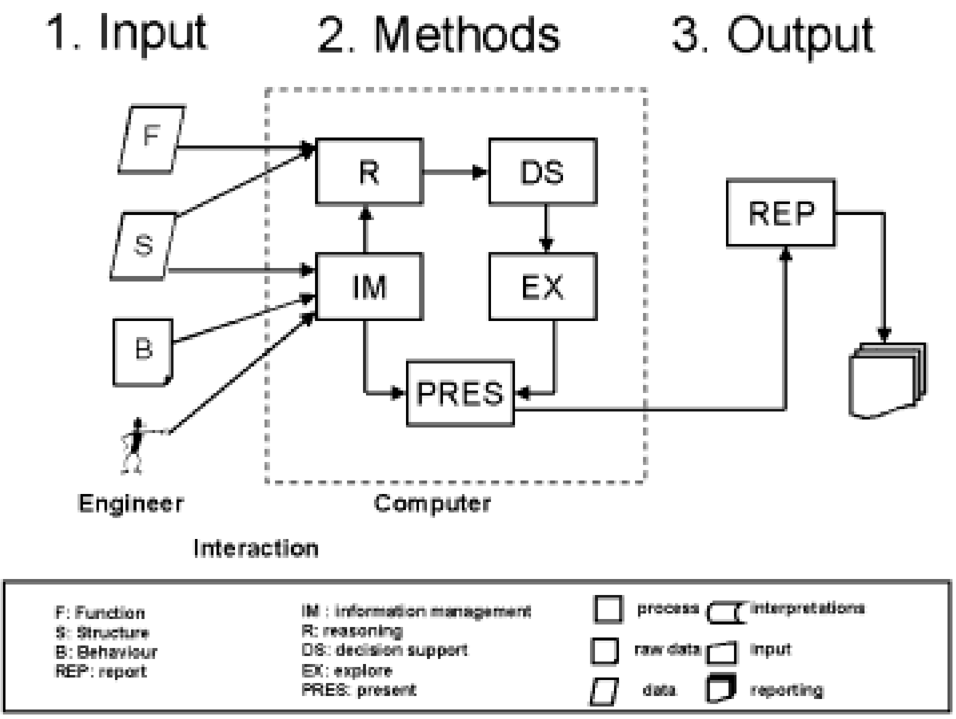

Figure 4: Function, Behaviour, and Structure is input into the iterative methods of information management, reasoning, decision support, exploration and presentation. Output is presenting and reporting in order to communicate observations related to structural behaviour

3.1. Input

This input describes the geometric aspects of the measured structure, the organisation and topology of the SOFO apparatus (Structure), and measurement data (Behaviour) and information related to the function of the structure. SOFO measures local deformations, and with sufficient sensors, information can be extrapolated for the whole structure.

3.2. Methods

3.2.1. Information management

This module formats the input for use. Filtering the data is important, as some measurements may be false, for example, measurements which represent physically impossible structural behaviour. If these measurements are not removed, data interpretations may not be useful. Each data point represents an area of the structure which has specific behaviour according to the seasons. If some data points are not within certain behavioural bounds, either the data set is unsuitable and the set can be thrown away, or there is a problem with the structure. The two alternatives call for the measurements to be retaken. Each data set is a potential description of a structure and each structure is unique in its behaviour, therefore a limited approach is taken to “cleaning” the data. Otherwise the data is no longer representative of structural behaviour.

3.2.2. Reasoning

The engineer is looking for a mathematical interpretation of the data set. This is done through Model-Based Reasoning (MBR). A model is a potential curve fit, and the data points are fitted to the curve. Due to experience and knowledge of the structure in question, the engineer may know which model summarises best a given data set (measured local deformations). The base model was developed in [13] and this module contains more models of varying levels of granularity. This is due to the uncertainty of a given model’s parameters therefore, an engineer has a choice of models. MBR is only as good as the model itself [14] consequently, several models are better than one. The data itself may indicate which model is the correct one to choose and create a convergence on the best model for better maintenance decisions.

3.2.3. Decision support

Decision support involves the comparison of alternatives. Decision makers are supported through the determination and display of an estimate of the range of possible interpretations. The interpretation of data through model selection is an abductive process. The engineer looks at all interpretations and decides upon the most suitable or a recommendation is requested and the system will indicate a best fit. This is currently performed through comparing the square of the sum of errors of each fit.

3.2.4. Exploration

The currently calculated behaviour is compared with a theoretical behaviour or a previous measured behaviour. Other manipulations are undertaken through a choice of viewpoints, where a request is made for certain data points (measured deformations) to be interpreted. This leads to multiple sub-interpretations in order to explore various structural behaviours whilst the engineer looks at the interpretations at intermediate stages. Undo and redo facilities aid the engineer in the exploration process.

3.2.5. Present

By using a computer, engineers want to gain insight into the problem and this does not necessarily mean a desire for more numbers [15]. Graphical exploration of information is a complementary to data analysis and reasoning. Some visualisations suggest hypotheses for further investigations and experiment and the presentation may aid in communicating results. Therefore, scatter plots, curves, and tables are presented to the engineer along with mathematical formulae and the graphical language of structural engineering (Figure 1).

3.3. Output

Reporting is an important part of structural monitoring therefore, screen output is saved in GIF and postscript format for inclusion in reports.

3.4. Toolkit Flexibility

The toolkit is implemented so that the modules can be used in any combination and in any order as shown in Figure 4. Each module has the same input and output of raw data, interpreted data, and engineer interaction, as shown in Figure 5. The only output to file is in the report module and combinations only need to use the reporting module if output is needed. The toolkit is to be used by a professional and therefore, the quality of the output of each module is not monitored. Furthermore, there are no critiquing facilities and no user performance monitoring.

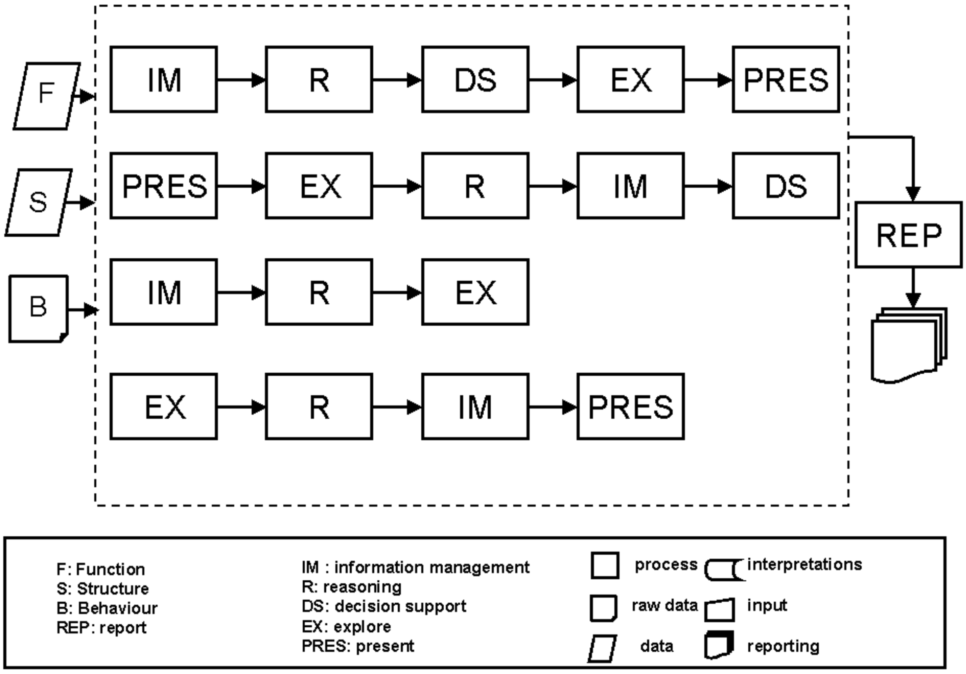

Figure 4: Software components can be put together in numerous configurations in order to provide flexibility. Each line in the dotted box represents an example configuration. The third line shows a configuration where input is entered; the input is goes through information management and is then reasoned and explored. In the fourth line, the information is explored, then reasoned with information management takes place, after which it is presented.

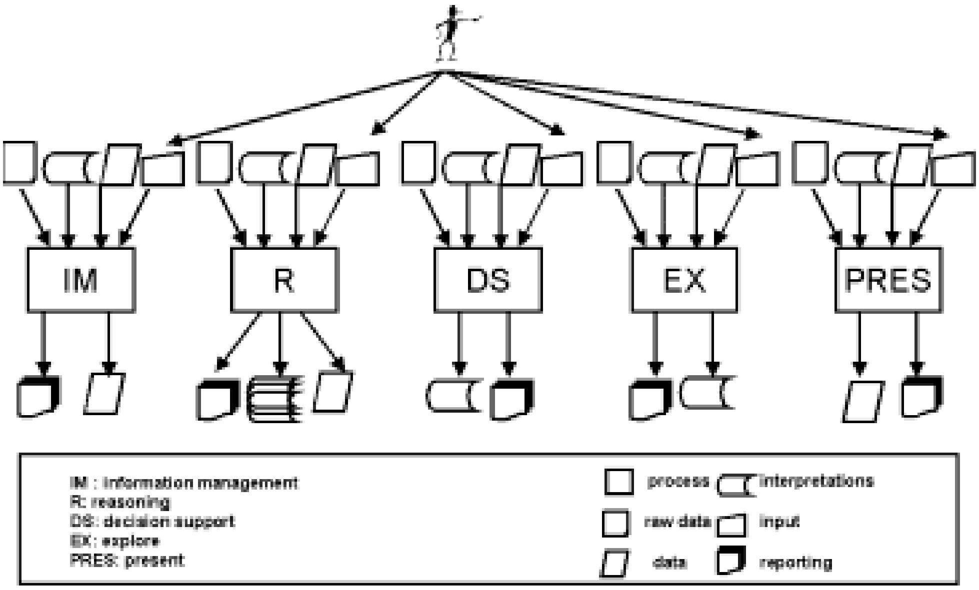

Figure 5: The software components have the same input and output from each module. Only the relevant output from each module is shown here

4. Evaluation

The evaluation of this software is ongoing. This section draws attention to the advantages of using this toolkit in terms of each software module.

Information Management: normally an engineer will do this by hand and depending on the size of the data set, it takes at least 2-3 hours. Reasoning: only one interpretation is performed therefore, gains are made here in additional interpretations, which may lead to more precise representations of structural behaviour.

Decision support and exploration are additional tasks, which would not necessarily be performed by an engineer due to time constraints. The decision support may lead to avoidance of error and the exploration to previously unanticipated behaviours.

Presenting: The integration of software and output of graphs and charts saves enormous time. Currently engineers change from one software to another in order to present adequately the information through graphics software. This toolkit provides a consistency in task execution and this leads to more tasks being undertaken, thereby resulting in error reduction.

5. Concluding Remarks

Replacing the word, human in Human-Computer Interaction with the word engineer, enables a development of software which is compatible with the engineer while remaining generic enough for all engineering tasks. This leads to a flexible, working environment for the engineer. This framework improves the speed, quality, and efficiency of task execution. Implementation of this framework in a toolkit of interchangeable modules shows much promise for providing increased support.

Future work will involved applying the toolkit to other engineering tasks.

Acknowledgements

This work is supported by the Swiss Commission for Technology, and Innovation, Logitech SA, and Silicon Graphics Incorporated. The authors would like to thank Iain Stalker, Simon Bailey, Pascal Kronenberg, Sandra Lloret and Benny Raphael for their input.

References

1. Goodier A.; Matthews S.: “Knowledge-Based Systems applied to Real-time Structural Monitoring”, Information Processing in Civil and Structural Engineering Design, Civil Comp Press, 1996, 263--270
2. Davis J.; Vann A.: “Monitoring Instrumentation Fault Diagnosis and Data Interpretation” Artificial Intelligence in Engineering, 1995, 923—938
3. Smith I.F.C.: “Abductive Engineering Support”, Computing in Civil and Building Engineering Press, TechnoPress, 1997, 1361--1370
4. ACM SIGCHI 1992 6.
5. Wegner P.: ``Why interaction is more powerful than algorithms'', Communications of the ACM, 1997, (40), 5, 80—91
6. Burr A.; Craig B.A.; Chung C.; So T.; Deveau T.; Shirer D.: ``Programs for Statistics and Data Analysis'' Computing in Science and Engineering January-February 1999, 16--21.
7. Hannus M.: http://www.vtt.fi/cic/hannus/islands.html
8. Anderson R.: ”Ethnography in Systems Design”, Human-Computer Interaction, 1994, Volume 9, 152—182
9. Salvaneschi P.; Cadei M.; Lazzari M.: “Applying AI to Structural Safety Monitoring and Evaluations”, AI in Civil and Structural Engineering, IEEE Expert, Intelligent Systems and their Applications, August 1996, 24—34
10. Sweeney R.A.P.; Oommen G.; Hoat L.: “Impact of Site Measurements on the Evaluations of Steel Railway Bridges”, 1997, IABSE Workshop Lausanne, 139—147
11. Gero, J.S.: “ Design prototypes: a knowledge representation schema for design”, 1990, AI Magazine, 11,4, 26—48
12. Inaudi D.: ``PhD Thesis Fiber Optic Sensor Network for the Monitoring of Civil Engineering Structures'', Swiss Federal Institute of Technology, 1997
13. Vurpillot S.; Inaudi D.; Scano A.: ``Mathematical model for the determination of the vertical displacement from internal horizontal measurements of a bridge'' Smart Structures and Materials SPIE Vol. 2719-05, 1996
14. Davis R.; Hamscher W.: ``Exploring Artificial Intelligence: Survey Talks from the National Conferences on Artificial Intelligence'' Chapter 8: Model-based Reasoning: Troubleshooting ed. Howard E. Schrobe, 1988, 297—346
15. Hamming R.: “Numerical Methods for Scientists and Engineers” Mc-Graw-Hill 1962