The application of ubiquitous multimodal synchronous data capture in CAD

Sivanathan, Aparajithan; Lim, Theodore; Ritchie, James Millar; Sung, Raymond; Kosmadoudi, Zoe; Liu, Ying

The application of ubiquitous multimodal synchronous data capture in CAD

Aparajithan Sivanathan *, Theodore Lim, James Ritchie, Raymond Sung, Zoe Kosmadoudi, Ying Liu

Heriot-Watt University, UK

HIGHLIGHTS

- A generic ubiquitous data capture framework is demonstrated via two case studies.
- Variety of inputs, interactions, biophysical data and design solutions are captured.
- Tight temporal synchronisation with commodity data logging tools is achieved.
- Demonstrates engineering knowledge capture linking CAD and PLM via generated metadata.
- The framework's use in future CAD and PLM systems is extrapolated.

ARTICLE INFO

ABSTRACT

Design is an interactive and iterative process where the designer's skills and knowledge are fused with emotive rationales aided by design tools. A design solution is thus influenced by the designer's creativity, experience and emotional perception. Consequently, there is a need within computer aided design (CAD) research for ubiquitous tools to capture the affective states of engineers during design activities to further understand the product design process.

This paper proposes a generic framework for ubiquitous multimodal synchronous data capture, based around the capture of CAD system activities, to monitor and log a variety of inputs, interactions, biophysical data and design solutions with a view to providing meta and chronological performance data for post design task analysis. The framework has been employed in two use cases namely, a CAD station activity and a collaborative design review. The results of these trials validated the architecture and use of the ubiquitous data capture approach demonstrating the practical application of time-phased data capture, analysis and the subsequent output of metadata in CAD environments providing a new perspective on, and a new way of investigating CAD-based design activities.

This research also extrapolates the framework's usefulness into future CAD and PLM systems by arguing why and how they need to adopt such ubiquitous platforms. It also subjectively points to potential opportunities and issues that might arise when implementing the ubiquitous multimodal metadata architecture in a real-life environment.

1. Introduction

Product design can be a complex and iterative process where the knowledge, experience, and skills of a designer/engineer are articulated through the design tools. The reasoning given by the designer behind every action performed in a design activity is highly complex and comprises inferred information, explicit and tacit knowledge, emotive communication and transient exposures to the subject matter [1]. Despite the associated product constraints, a designer's knowledge and rationale play a significant role in the design. These are fuzzy by nature, difficult to capture and therefore difficult to document. Although self-reporting at each design phase can be applied, it is heavily dependent on the memory of the designer, open to interpretation and also creates interruption during the design process itself.

Present-day computer aided design (CAD) solutions can produce a history of the designed solution within the internal bounds of their software package, e.g. revision history of a CAD part, tracking of changes made to a solution, etc. However they miss out the valuable information about external endeavours particularly those that happen during the design activity such as dialogues, sketches,

* Correspondence to: Department of Mechanical Engineering, Heriot-Watt University, Edinburgh, EH14 4AS, UK. Tel.: +44 0 131 451 4306.

E-mail addresses: a.sivanathan@hw.ac.uk (A. Sivanathan), t.lim@hw.ac.uk (T. Lim), j.m.ritchie@hw.ac.uk (J. Ritchie), r.c.w.sung@hw.ac.uk (R. Sung), z.kosmadoudi@hw.ac.uk (Z. Kosmadoudi), ying.liu@hw.ac.uk (Y. Liu).

notes, informative directives, knowledge and the emotional states of the designer, as well as other physical activities which influence the design process.

The literature demonstrates that there is a whole host of data required to be monitored and logged to enable design tasks in computer aided environments to be revisited, interrogated, analysed and understood [2,3]. The future of design task analysis and knowledge/information capture lies in studies that involve the logging and analysis of formal computer aided engineering data, e.g. specifications, geometries, properties, calculations, simulations, in conjunction with the multimodal information about the activities performed during the design process. In general, multimodal interaction capture involves capturing audio-visuals, document access history, inter-personal interactions and psycho-physiological data in a ubiquitous, time-phased manner. As a consequence of this, there is a requirement for standard protocols which researchers can use to implement such a monitoring methodology as a tool to take forward the next generation of design research.

This paper proposes a generic framework for ubiquitous multimodal synchronous data capture, based around the capture of CAD system activities, which was employed to monitor and log a variety of inputs, interactions, psycho-physiological data and design solutions with a view to providing meta and chronological performance data for post task analysis. The framework predominantly addresses a technical solution for the ubiquitous and holistic capture of design activity. Through this work it has been demonstrated that it is possible to build up an end-to-end pipeline which ubiquitously captures the design activity and represents the captured meta-information against CAD records in a time-phased manner.

1.1. CAD—the case for unobtrusive ubiquitous design data capture

Jin and Ishino [4] proposed a design activity knowledge acquisition (DAKA) framework to extract the design activity knowledge by capturing the designers' design moves and used a function-based design operation-mining algorithm to extract meaningful design operation sequences. DAKA established the product model road map that represents the trajectory designers walked through during the design process by capturing the sequence of events occurring while interacting with CAD software.

Besides the CAD software and CAD models, a designer would also interact with other media such as text, audio, video and sketches. Shipman and McCall [5] proposed an integrated approach to capturing design rationale and associated design communications by means of the 'hypermedia'. Two systems, PHIDIAS and Hyper Object Substrate (HOS), were used to capture and integrate a variety of hypermedia related to the design process and to structure the unstructured hypermedia information. A vector-based node-link structure was used to represent the connectivity between hypermedia, serving as the basis for design rationale. Although the structuring of hypermedia is studied, no temporal records were utilised in this work to relate them with the chronological performance of the design process.

Design activities are dynamic and any information resources accessed during a design session can be captured and potentially reused in future design tasks. However, structuring design knowledge, e.g. inferred, tacit, explicit and transitive, in a machine accessible format for storage and retrieval for computer-based knowledge support is not trivial. Campbell et al. [6,7] explored a methodology for profiling computer-based designed activities relative to the order and timing of resources accessed during a design session, where the data is captured along with the creation of formal design representations, e.g. design drawings, reports, etc., based on Bayesian inference. They then tried to relate it to the designer's focus or goal which was interpreted by the actions

observed at the computer interface, e.g. keyboard input and interaction with the CAD software, so that a relationship between documents based on the context of their usage can be identified.

The Knowledge Enhanced Notes (KEN) system introduced by Conway et al. captures collaborative activities in a design meeting and generates an enhanced documented record of the meeting with a view of re-using at later stages in the product lifecycle. In addition to the discussions and decisions made during the meeting, KEN records the resources accessed and the temporal information about them [2]. The benefits of the multimodal information have been generally realised specifically as informal data about the design process in addition to formal design records, which enables revisiting and understanding the design process in product's extended lifecycles [8,9].

Revisiting and interrogating the design process with regard to the multimodal metadata requires a pristine link between the formal data and informal data. Generally, this has been overlooked and the informal data searchable or not-searchable on its own will not have the associativity with the formal data. Therefore the captured metadata should be associated with the part name or a product lifecycle management (PLM) identification property in order to provide the bidirectional accessibility. Preferably, the best time to capture this link is during the actual activity, when the formal design records are being created or accessed and the informal data is being captured.

Analogous to CAD, Rea et al. [10] used BAMZOOKi, an intuitive 3D design environment, to study design activities. This work specifically enabled the automatic capture of the design process via a customised BAMZOOKi interface and generated log files based on extensible mark-up language (XML) and Process Specification Language (PSL) during the design activity. The log files were subsequently processed to output various representations of design knowledge. Integrated computer-aided manufacturing DEFinitions IDEF0 [11] and Design Rationale Editor DRed [12] diagrams were automatically generated to formally represent the designer's activity and design rationale. English-syntax instructions, annotated video clips and a design storyboard, were generated to represent the design knowledge and processes. Similarly design activities in a virtual reality based cable organisation system were also studied by capturing user activities identical to the CAD design process [13,14]. This further demonstrates and validates that design activities can be potentially transformed into understandable and CAD-neutral format knowledge representations.

1.2. Capturing user activity

Comparable to analysing a CAD activity, capturing user activities is common in Human Computer Interaction (HCI) studies. Usability studies for investigating software application user interface (UI) extensively used user activity monitoring systems [15–17]. A survey by Hilbert and Redmiles [18] classified and described a wide range of techniques used in HCI research for extracting usability information from user interfaces. They categorised which usability indicators were analysed, e.g. user behaviour, performance, cognition, attitude, stress level, and which multimodal data collection techniques such as UI events, audio, video, psychophysical recording and subjective measures such as interviews and surveys, were used to study the usability. This work also highlighted that user activities can be classified and analysed into different levels according to their level of granularity and abstraction; the abstracted activities can be inferred by interpreting a stream of events related to the low-level interactions [18].

Fig. 1 illustrates examples of CAD user activities during a typical CAD task, approximately representing the variation from low-level interactions to inferred abstract tasks. A balance between the granularity of the data capture and the level of inference needed to
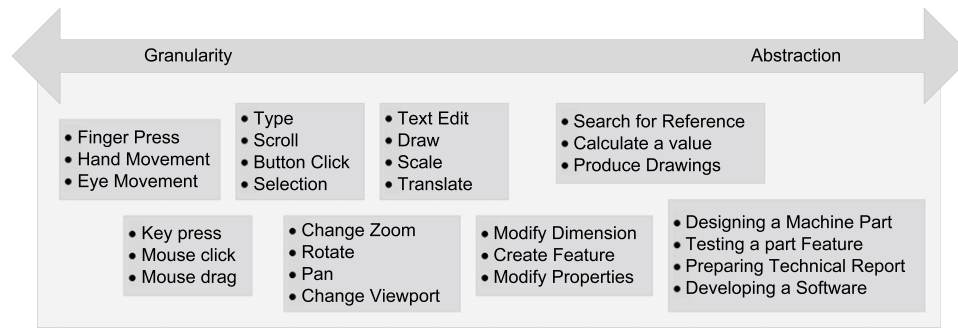
Fig. 1. Example activities in a CAD task approximately illustrating the granularity of the user activity and the abstraction of tasks.

incorporate the abstraction should be maintained while capturing and analysing user activities [19]. Accordingly, adequate low-level, granular activity data should be captured to recognise reasonably abstract tasks [16]. Too much data, when combined with extraneous information, create difficulties in drawing inferences about the task and can demand high computational and storage resources.

Standard operations in a PC, such as file operations, text manipulation, accessing the web, voice calls, etc., are often monitored to analyse the user activity [17]. Capturing events automatically generated by application, instead of low-level keyboard and mouse interactions, is a commonly used technique in user activity monitoring systems. These events are usually pre-determined and the applications customised to generate these events automatically while the user interacts with the system [19,20]. Events are chosen and organised based around the domain of study, for example a study about document interactions considers file system events, internet browser events, and application window events [6].

Software tools are normally required to capture and exchange event information across the UI application and monitoring software. Microsoft component object models (COM) and dynamically linked libraries (DLL) are often used to build the infrastructure for receiving event notifications in Microsoft applications [19,20,17]. Intercepting messages invoked by the operating system for keyboard and mouse events is another commonly used technique for monitoring user activity [19,21].

Events raised by the application or the user interface represent instantaneous activities occurring for a discrete period of time. In contrast, user activities can be observed in a continuous manner, e.g. video and audio recording, screen capture, and continuous mouse movements [15,22]. Psycho-physiological measures are increasingly being employed to establish a better understanding of the internal state of the user. Heart rate/electrocardiography (EKG), facial muscle activities, eye tracking, pupillometry, electroencephalography (EEG), electromyography (EMG), and galvanic skin response (GSR) are some of the commonly used biometric measures to monitor user activity [15,23–27].

User activity monitoring systems have largely focused on logging events occurring between the user and the UI. However, the recording user activity in a task-based situation creates more challenges. A design task involves human factors beyond examining the interaction solely between a designer and software. Personal aspects such as knowledge, experience, creativity and emotional perception do influence the design solution. Consequently, psycho-physiological data provides an alternative for analysing affective design activities. Complementary to UI event logging, which provides an explicit form of information about the user interaction within the UI, the human-centric data represents an implicit form of user interaction. A need for a common set of tools to integrate and build up a user activity monitoring system has been noticeably identified in the past [28]; however, this work is the first attempt at specifying, designing and implementing such a generic framework for CAD.

1.3. Requirements for a ubiquitous framework

The case for using informal metadata relating to the design process in addition to formal design records has been described and consequently the requirement of multimodal capture is established throughout the Sections 1.1 and 1.2. The unobtrusiveness in the design process necessitates a ubiquitous and dynamic system that operates in such a manner that the capture tools are embedded into the working environment; therefore it does not create an interruption of or add extra workload to the on-going process. The use of ubiquitous, multiple modalities necessitate tight temporal synchronisation tolerances between captured data streams in order that a meaningful understanding of the CAD activity relationships can be established. For this reason systematic, implementable arrangements are presented that accommodates a wide range of data capture tools and blends them seamlessly into the environment and the process. It particularly addresses a technical solution for consistently extendable, ubiquitous metadata capture systems.

The rest of paper is organised as follows. Section 2 presents a generic ubiquitous framework detailing the implemented key modules and elements. The way components of the framework are implemented to solve issues as they arise in a ubiquitous capture environment is described. Section 3 provides two use cases, i.e. a CAD station and a collaborative design review system which are described to demonstrate how the generic ubiquitous framework was employed to monitor and log a variety of multimodal information during design activities. As an outcome of applying the framework, a variety of inputs, interactions and designer biophysical data are represented with a view to providing meta and chronological performance data during a CAD task. It has been also demonstrated that a design task can be captured and analysed in minute detail with the aid of various captured data representations.

The appropriateness of the framework is extrapolated into its use in future CAD and PLM solutions in the Section 5. Based on the previously described use cases, two possible configurations of metadata collection systems are proposed. Various issues that could potentially arise when such a metadata collection system is built in a real-life design environment are also outlined. Section 6 summarises the key contributions to the field of ubiquitous synchronous data capture while pointing towards target users of the framework in engineering and other fields.

2. Ubiquitous capture framework

The ubiquitous capture framework is presented in abstract terms, whereas C++ and C# are used to develop the actual data capture system. The abstractness ensures that it is generic in form and is maintained throughout the actual implementation by the use of object oriented programming concepts such as classes and inheritance. Integrating multiple capture devices, handling data flow between components and synchronisation, are the main issues addressed by the framework.

2.1. Integrating multiple capture devices

Multimodal data capture tools include both data logging software and hardware-based capture devices. Hardware-based capture devices are usually used in combination with an application programming interface (API) or Software Development Kit (SDK) where the API/SDK provides an interface to hardware control and access to the captured data. For example, functions for initialising the device, releasing the device, starting data capture, stopping data capture and configuring parameters are typically available control functions in an API. Although the actual tasks performed by control functions are often similar, naming conventions and ways to access these functions often vary significantly across different device manufactures [29,30]. A possible reason is attributed to different connection interfaces, e.g. USB, Serial, Ethernet, etc., and protocols, e.g. USB, TCP, UDP, RS232, etc. Although the term 'devices' generally represents hardware-based data capture tools, this term also refers to software-based data logging tools, e.g. CAD software, unless otherwise stated in this work.

A unified access interface to support the ubiquitous nature of the framework has been defined such that all the multiple capture devices' control functions can be accessed homogeneously. Generally, a wrapper interface is implemented which hides the device-specific function naming conventions and merges them if there are multiple function calls required to accomplish one specific function, i.e. loading API DLLs, initiating connections with the hardware and checking for device availability that can be merged into the 'initialisation'. Principally the unified access interface abstracts device-specific function calls for data logging into a universal interface. Primary functions in this are as follows:

- *Initialise* — initiating communication with the device and configuration; thereafter it is ready to be used.
- *Release* — freeing the device and the API.
- *Start* — starting the actual data capture.
- *Stop* — stopping the capture.
- *Reset* — resetting the device in case errors have occurred.

Conversely, some features are unique for particular types of device only. For example, selecting the input channel, e.g. HDMI, DVI, and Composite, is an exclusive feature for video frame grabber devices only. Therefore, the framework allows access to these functions directly, bypassing the unified access interface. It is usually preferable to access device-specific features through a bespoke device-specific interface while accessing the generic functions through the unified interface. This enables multiple devices to be controlled via a centralised control panel without losing the diminutive control of the device. Fig. 2 illustrates an example unified access interface and centralised control panel.

The conventional method of building a data logging arrangement is to integrate every device into one single large program. This creates potential complications in a ubiquitous data capture system, especially when the data capture configuration changes, i.e. when a device is not available or a new device is added to the system. In this case a programmer would usually modify, duplicate or rewrite the code from the previous setup for the new configuration. The concept of a unified interface implemented in the framework allows devices to be used in a modular manner in a variety of configurations without rewriting or duplicating the code. An initial implementation of the unified interface for a device is always needed after which it can be reused to handle alternative multiple data logging arrangements without any alterations.

The unified interface also allows devices to run independent of each other. It enables devices to attach and detach at runtime, permitting the ubiquitous data capture system to change dynamically. This unique feature embedded within the framework is particularly useful since it means that the operation of the system will not be disturbed if, say, a device fails or needs to be removed, when a new device is accessible to join the system, and where modifications in runtime are required. This is achieved by allocating a status parameter to every device which can be checked by the centralised control panel to determine the availability or status of a device. The key status variables are:

1. *Dormant* — device is not available.
2. *Not-initialised* — device is present but not ready for use.
3. *Ready* — device is available for use.
4. *Running* — device is capturing data.
5. *Dead* — device is failed, unknown state.

2.2. Data types and real time access

Each of the capture tools within the ubiquitous framework produces a variety of data types depending on the captured content, e.g. discrete key press, mouse movement coordinates, video, location-based multichannel physiological measurement such as EEG, etc. The characteristics of these multimodal data vary extensively based on the format of the data, its resolution and frequency. Table 1 explains the mixture of types of data produced by devices used in CAD environment experiment in this study. Care should be taken to handle those dissimilar data types, for example, the way the 19 channel EEG data is treated, which corresponds to different locations on the head, is different than, say, 2-dimensional pixel based screen capture. Data buffers involved in retrieving the data and the processing algorithms used also vary.

In addition to the dissimilarity in data formats, the method by which they are captured is also important. Several data channels are embedded within a stream, i.e. continuously at a constant frequency, e.g. EEG, eye tracking, video, while the others are captured as discrete events occurring in the environment, e.g. CAD activity logging, key presses. Remarkably, some of the data can be captured in both streams and in an event-based manner, e.g. mouse activity can be monitored for discrete clicks while tracking the cursor movement continuously, and tracking the eyes continuously while triggering specific events when the user looks at a specific region of interest on the screen.

The bandwidth and frequency of a data channel are important factors in deciding how it should be handled. Biophysical signals, e.g. EEG, GSR, are captured at high frequency (2048 Hz) from USB hardware. There is the potential for missing data packets from devices if they have not been read at the required frequency. It is crucial to allocate appropriate buffer sizes and also remove the data from the buffers at an adequate rate. HD video capture needs handling vast amounts of data (1280×720 pixels $\times$ 3 byte colour depth X 30 fps = 79.10 Mbytes per second) and potentially creates a bandwidth bottleneck within the data capture system. It is common to use a video codec to compress the video in real time; on the other hand runtime video compression puts high demand on the CPU. Although CAD logging is normally not a computationally demanding task, the events triggered by the CAD software are sporadic since it is based upon user activity, e.g. the user spins the model and changes the view zoom occasionally. This produces varying amounts of data in short but intensive bursts, producing a non-uniform load on computational resources which might cause issues in the real-time scheduling of tasks involved in the system.

Data handling modules are implemented in the ubiquitous capture framework to address issues related to multimodal data. Abstract and modular-based implementations of stream type and event type data are built into the framework. A programmer may use these abstract modules as a skeleton to implement personalised data handling modules for different data types in a customised data capture system. Fig. 3 shows various configurations of how the data handling units in the ubiquitous data capture environment can be linked using the appropriate modules. Brief descriptions of each configuration are now given.
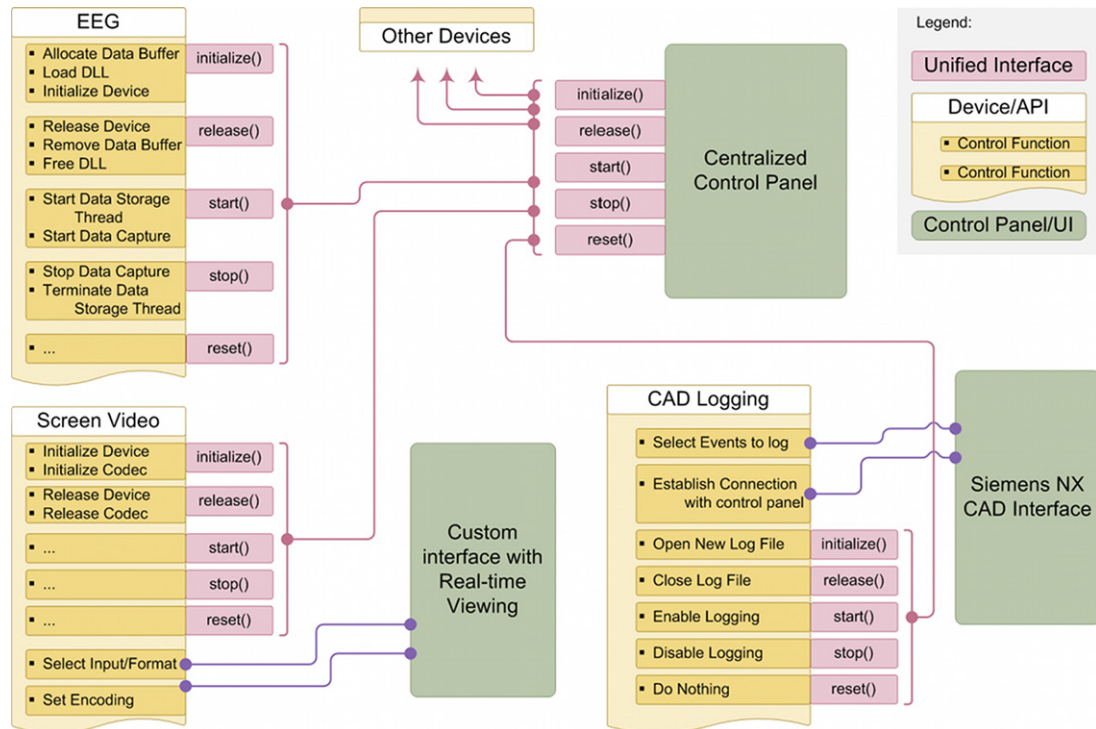
Fig. 2. An illustration of a unified interface with centralised and device-specific control panels. A part of the system and example functionalities are used to describe the concept.
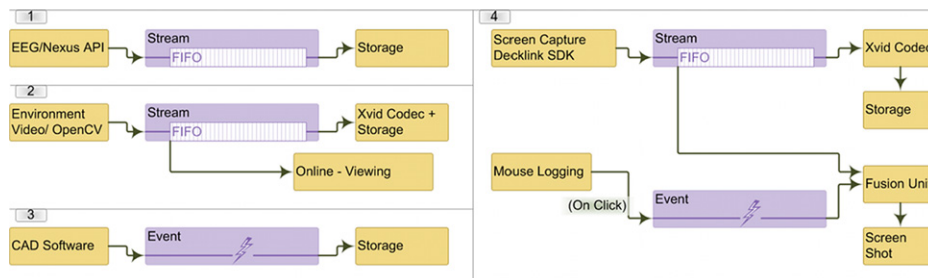

Fig. 3. Four examples of data flow configurations for stream- and event-based data.

Table 1
Multimodal data captured by different devices listed along with the device/API, data format, frequency and type of data.

| Data	Device/API	Stream	Event	Data type	Frequency (Hz)	Description
CAD logging	Siemens NX/UG open API		×	String	–	Event description generated by customised UI
Spread Sheet logging	Microsoft excel/VBA		×	String	–	Excel spread sheets events
EEG	Nexus 32/API	×		$19 \times$ float	2048	$19 \times$ EEG channels corresponding different locations on the head
GSR, EMG, Heart rate	Nexus 32/API	×		$1 \times$ float	2048	$1 \times$ bipolar channels each sensor
Eye tracking	Tobii X50/clearView API	×	×	$4 \times$ long $12 \times$ float	50	Left and right eye – gaze – X, Y, pupil diameter, distance, validity
Key press	Keyboard/windows hook		×	$1 \times$ integer	–	Key code
Mouse	Mouse/windows hook	×	×	$2 \times$ integer $1 \times$ byte $1 \times$ integer	–	X,Y cursor position button click, wheel
Screen video	Intensity shuttle/ DeckLink SDK	×	×	$1280 \times 720 \times 3$ byte	30	720p hi definition (HD) screen capture, video/snap shot
Environment video	Sony playstation eye/Open CV	×		$640 \times 480 \times 3$ byte	30	VGA video capture
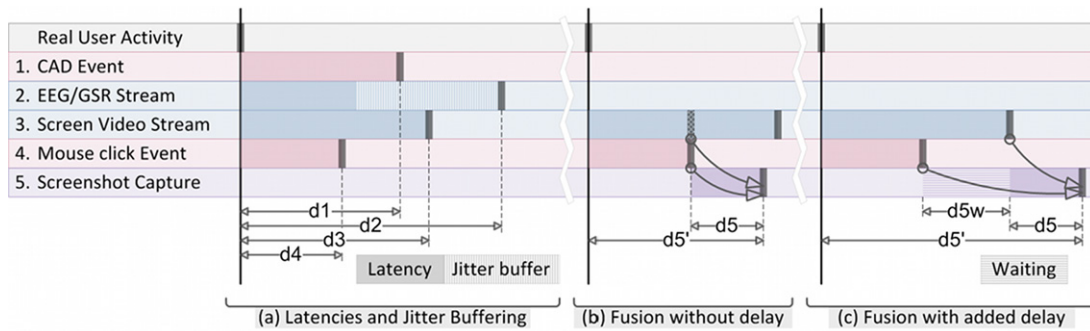
Fig. 4. Illustration of latencies in data channels with respect to instantaneous user activity. (a) Fusion units, i.e. screenshot capture on mouse clicks, are influenced by latencies. Offsets in the streams create skew (b) in the generated screenshots. This can be compensated by adding an extra delay (c) as required. d1–d5 = latencies in channels, d5w = added waiting time, d5′ = overall latency for the screenshot.

1. First in first out (FIFO) buffers are extensively used for sharing data from one unit to the other, where FIFO buffers support exchanging data between multiple execution threads. Multi-threading provides better handling between time critical tasks and long running background tasks. Collecting data from EEG hardware is a time critical high priority task. Therefore a high priority, high frequency task can collect EEG samples from the hardware and store it in the FIFO, whereas a long running storage task can transfer the data in batches from the FIFO to a disk storage. An abstract stream data module has been implemented in the framework and say it can be extended to support EEG data formats. Correspondingly, the same module was extended to support associated stream-based data types in the following configurations.

2. It is reasonable to have a ubiquitous data capture system which has a real-time viewing capability which can be used for inspecting the captured data online. The video from any environment camera, capturing user activity viewed online while the recorded video frames are compressed by the codec and stored in a video file. However, online viewing is only a supplementary feature and has less importance than other components in any data capture system. It is good enough to miss video frames for viewing and hence appropriate to run a low frequency, low priority thread with flexible timing requirements. A feature has been implemented to retrieve samples from the FIFO without removing the samples from the buffer so that every frame can be securely collected by the storage thread.

3. In contrast to the stream module, an event-based module is implemented in the framework which deals in particular with sporadic data. Events triggered in CAD software are passed through this event module and can be linked to the other system units if necessary.

4. Two functions, namely capturing the user screen video and generating a screen shot when a mouse click event is fired, are combined in this configuration. A fusion unit was employed to retrieve a frame from the user screen video stream whenever a mouse click event is received.

Data handling modules implemented in the framework are designed to be lightweight and high performance. In contrast to other data logging systems [29] which upload the data to a centralised database, data have been handled by the framework in a distributed manner, therefore enabling high bandwidth and low latency data handling. Nevertheless, the captured data can be committed to a central repository using a low priority task if required.

2.3. Data stream synchronisation

Data samples collected via miscellaneous devices are routed through various data paths thus possessing disparate latencies.

These streams must be aligned to produce valid correlations and hence useable information can be generated. The validity of the inferred data produced by fusion units depends on the temporal alignment of the data fed into these data fusions. For example, the screen video, eye tracking and GSR must be temporally synchronised to derive a reasonable conclusion about user's emotions in an instant [31]. It is also important to synchronise the events occurring in the CAD environment, e.g. events triggered by the CAD software, mouse clicks, with the data streams such as biophysical signals and video. An illustration of the potential latencies involved in data channels is shown in Fig. 4.

Assorted devices running on isolated platforms with physically separated hardware and clocks essentially run asynchronously. It is normal to observe jitter while accessing the live data streams. Although devices might be internally capable of sampling signals at sufficiently precise intervals, general purpose PC operating systems and connectivity solutions, i.e. USB, TCP, create non-deterministic distortions in the synchronisation of the data streams [32,33]. Commodity devices do not often provide advanced clock/stream synchronisation methods. Nevertheless, it is not pragmatic to expect a common synchronisation technique in a ubiquitous data capture environment where diverse and multimodal hardware is used. To address this, the framework incorporates a latency/offset compensation parameter for every data channel. The offset parameter is used in online-fusion units to compensate for the offsets in fused channels or in post-capture analysis to re-synchronise the channels. In the previous example in Fig. 3, configuration 4 is a fusion unit creating screenshots for mouse clicks where the screenshots are skewed by the difference in offsets in the data streams (Fig. 4b and c).

Stream and event type data handling modules described in Section 2.2 are equipped with the offset and jitter parameters where a suitable method to estimate them can be chosen by the programmer, i.e. fix to a constant offset, programmatically choose the offset, online jitter monitoring, etc. Monitoring parameters online provides the information about synchronisation-related issues, e.g. clock drift in recording hardware, communication bandwidth fluctuations, etc.

2.4. Interconnection of distributed devices

The diversity of devices used (Table 1) in monitoring CAD environments often requires special considerations in choosing hardware and software platforms. Support is provided by various device vendors for different operating system platforms and maintaining issues caused by legacy software/hardware influences selecting the hardware and software platforms. Incompatibility between devices and platforms occasionally requires a mix of platforms. The framework supports interconnecting devices running
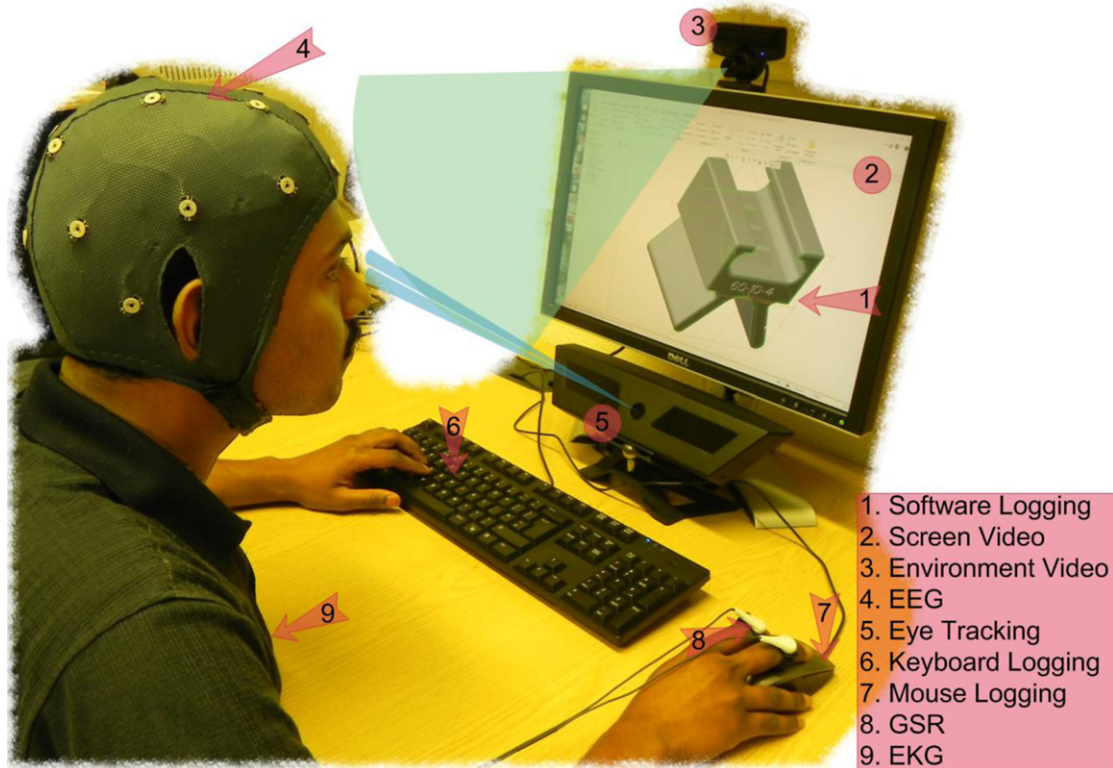
Fig. 5. Ubiquitous multimodal data capture framework used in a CAD design environment. Multiple capture devices used in the system are listed and highlighted.

independently in isolated platforms with the aid of a communication interlink, e.g. TCP, UART Messages, that relay the unified interface function calls and also the synchronisation related messages. Furthermore when the data streams or an event is exchanged between components of the framework, the latency in these interlinks also needs to be counted when synchronisation is considered.

3. Use case 1: evaluation of the framework in an engineering design activity

The experimental ubiquitous data capture system was setup to demonstrate the applicability of the generic ubiquitous framework to an engineering design task. This experimental system is also a case study for the generic ubiquitous framework and to test out how its components fulfil the requirements for capturing an engineering design activity. Previously described concepts and modules of the framework are implemented and tested by applying them to the experimental capture of design activity and thereafter justified by examining the captured data during the experiment. This section presents an outline of the experimental design task followed by descriptions of the framework implementation and how its modules were used to handle issues related to the design environment.

A design task to optimise a bracket to support the given pin force was to be trialled on an industry standard CAD package, i.e. Siemens NX^{TM}. A total of 24 engineering students were recruited to establish a record on their cognitive loads during design activities. There was no time restriction for the user to complete the task. The design task was divided into two stages:

(a) To modify the bracket such that it will not fail given an applied pin force. The student was required to calculate the allowable tensile and shear stress based on the bracket material and then, through the interface, configure the cross-section (breadth and thickness) of the bracket (Fig. 6).

(b) To calculate the tensile and shear stress including the weld type and sizing for the bracket.

In order to understand the affective states and the cognitive processes of the designer during the iterations of the given task, psycho-physiological measurements were recorded and synchronised with other modalities such as keyboard and mouse interactions, video recording, etc. (Fig. 5).

High temporal synchronisation was required in the recorded data, which included the EEG, EKG, eye movement, NX log file, mouse clicks and a video of user design activities. The EEG and ECG data are processed for emotions and pupil sizes were correlated to the cognitive load. By studying the psycho-physiological attributes associated with a CAD-based engineering design process, it is envisaged that interrelated engineering behaviours can be more deeply understood and, consequently, lead to more natural and intuitive CAD user interfaces.

3.1. Integration of ubiquitous devices in the CAD task

Standard Siemens NX^{TM}CAD software was customised using the UG Open API according to the unified interface structure as proposed in the framework. The customised module also generated markers for predefined actions, e.g. modifying the values of text boxes, pressing command buttons, pan and zoom-view manipulations, etc., carried out in the CAD software. This module was also responsible for regulating the other capture tools setup in the environment. For example metadata capture of other devices is started/stopped when the designer starts/stops working on the CAD, i.e. controlling the whole metadata environment.

Fig. 7 shows how the capture devices were setup and how three different workstation platforms were used in the ubiquitous data capture environment for the CAD task. A low latency system, PC1 (Windows 7, 32 bit), is used as the centralised control panel and for time stamping purposes. Simultaneously, it is also used with EEG and eye tracking devices, for which platform

Fig. 6. This figure illustrates the experimental task in the customised NX interface which allows performing of the stress calculations and shows the subsequent results.
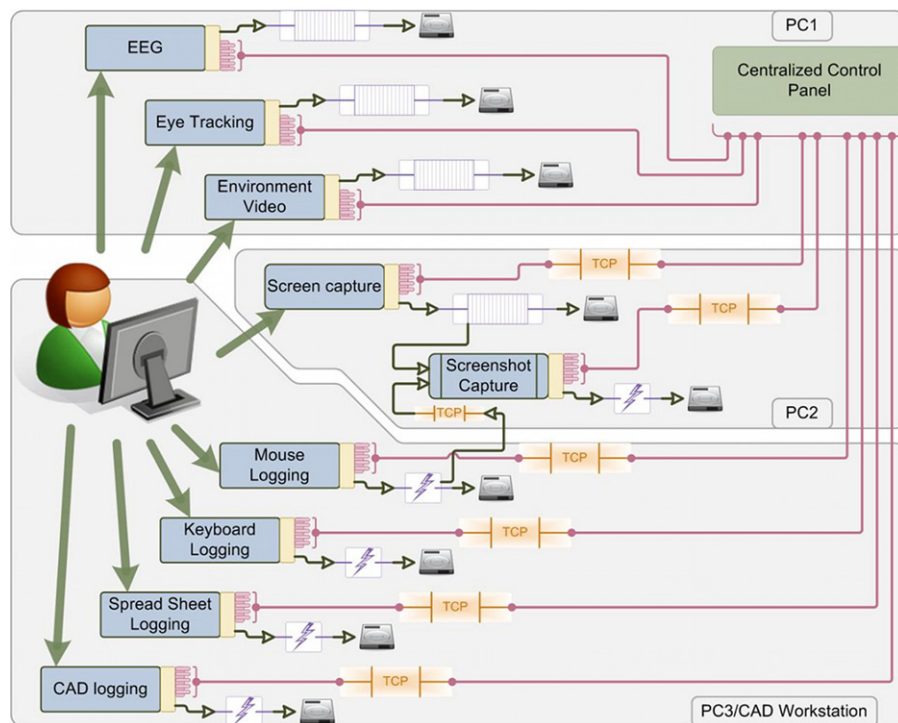

Fig. 7. Arrangement of device units showing how event and stream modules are used to handle the data flow and the unified interface is used to access device functions. TCP interlinks have been created to support interconnection of distributed devices.

drivers and APIs only support 32 bit platforms. PC2 (Windows 7, 64 bit) has been purposely tuned for high bandwidth and storage capabilities with RAID-0 disk storage and USB3 features. The CAD workstation (Windows XP, 32 bit) was built addressing the legacy support issues. TCP interlinks were implemented to interconnect distributed devices with the centralised control panel and to create connections between devices wherever the data flow between different distributed device units was required.

3.2. Handling latency and jitter in data channels

Mouse clicks and screen video stream are captured by different devices and associated APIs (Table 1). The offset between these channels creates an observable skew when these channels are fused together. Grabbed video frames and the mouse clicks

are analysed to validate synchronisation and the concept of latency/offset compensation. Although the offset compensation is applied for every channel, these two channels are chosen for analysis because the mouse pointer on the screen and the button click event can be manually observed in the video frames. Mouse x and y coordinates logged in a click event by the Windows hooks is correlated with the screen video frames. Fig. 8 shows the timeline of video frames, a mouse click event and the mouse click coordinates. The skew can be visually observed where the click coordinates and the mouse pointer do not match. When the offset compensation is applied the mouse pointer is then aligned with the logged coordinates in a subsequent frame and hence validating the synchronisation.

Monitoring the jitter and frame rate of a device in run-time provides more information about the runtime behaviour of the
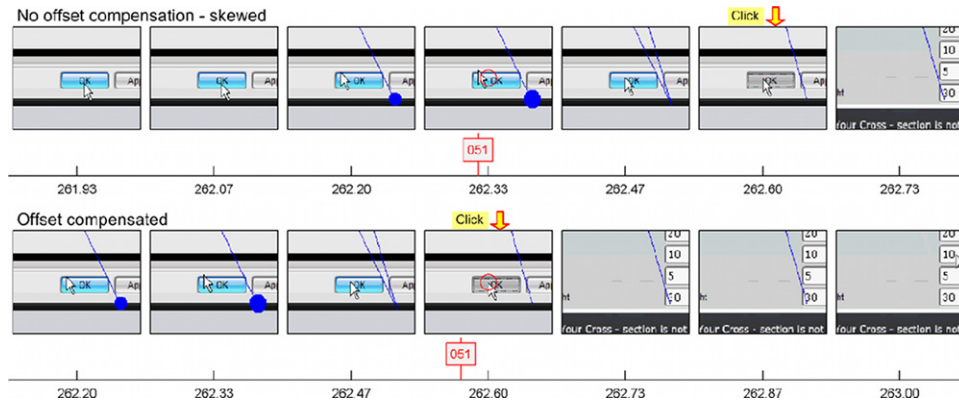
Fig. 8. Timeline of grabbed frames and mouse click event, where the logged mouse coordinates on a click event is superimposed (red) on the closest video frame. The skew can be visibly observed (top) before compensating for the offset and thereafter the click is synchronised (bottom) with the video stream after applying a compensation of 267 ms. Eye-gaze points are also overlaid on the frames (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
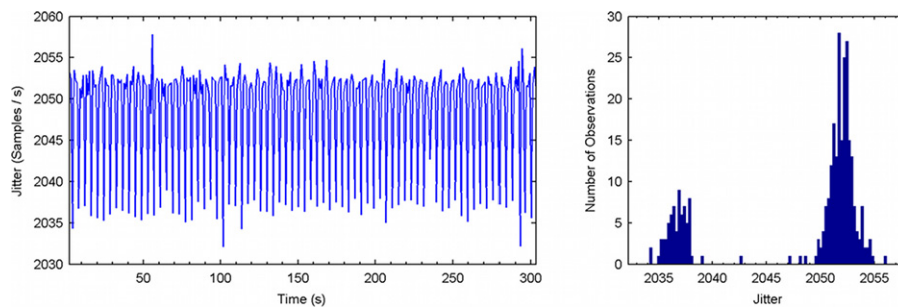

Fig. 9. Jitter observed in run-time for the bio-physical monitoring device, with the theoretical frequency of 2048 Hz. The number of collected samples has been recorded every second and the resultant sampling rate is calculated against the CPU's hardware high precision clock.

device and data stream generated by the device. This information is captured by the data handling modules implemented in the framework (Section 2.3) which provides a precise understanding about the dynamic characteristics of the synchronisation. For example, the Nexus, bio-physical monitoring device, samples signals at 2048 Hz according to the manufacturer's specification. The number of samples delivered by this device via the associated API is observed at fixed intervals. Subsequently the instantaneous sample rate was calculated using the time provided by a high precision hardware clock in the central processing unit (CPU). The continuously measured sample rate and the observation of jitter are illustrated in Fig. 9. The presence of jitter and the deviation in the sample rate in this data stream have been revealed using the data handling modules implemented in the framework. Similar distortions were observed in other devices, e.g. eye tracker, frame grabber which are possibly caused by the non-deterministic characteristics of the data paths, e.g. USB, Ethernet and the inaccuracies in task scheduling. Subsequently, FIFO buffers implemented in the framework were used to treat the jitter and offset compensations were applied to individual data channels so that they can be temporally aligned.

3.3. Bio-physical signals synchronised with CAD events

Events triggered by the CAD software while the engineer performs the design task, which are logged by the customised Siemens NX interface. These events were logged in a log file (Fig. 10) with a distinctive event identifier and the corresponding time; therefore they can be synchronised and correlated with other data channels. These event identifiers were used by event handling modules in the framework to handle them unambiguously.

A timeline of events triggered in the design environment and bio-physical signals are shown in Fig. 11. 19 channels of EEG

signals and EKG are captured using the bio-physical monitoring device while eye gaze and pupil radius are captured by the eye tracker. These independently running devices and CAD logging are integrated using the framework to function together as a ubiquitous capture system.

The eye blinks captured in the data can be used to justify the temporal alignment of integrated data streams. Eye blinks can easily be identified in the EEG data stream by the eye blink artefacts, while the eye tracker loses tracking of the eyes when the user blinks. Fig. 11 shows a tight temporal alignment of these data streams where EEG and eye tracking data can be visually examined for the blinks. This consequently substantiates the implemented data handling modules and synchronisation techniques of the framework.

3.4. Design activity representations

The time-phased log files recorded during the design activity were parsed by automatic tools [34] to produce various design activity representations. A modified DRed representation, which incorporates the time duration of tasks, is illustrated in Fig. 12. Time-phased captured EEG signals were automatically processed for emotions using a fuzzy model [35], was synchronised with the CAD events with the intention of analysing the engineers emotions during the design task. Fig. 13 illustrates that the processed emotion data is mapped onto a manually drawn IDEF0 diagram using the events recorded by the framework as synchronisation milestones.

3.5. Outcomes of the use case study

Various studies were performed in the current state-of-the-art research with the aim of understanding the affective status of

```
22   !!!&MACRO EVENT FOCUS_IN 1 0, 1900544, 0, 0, 0! (Application CB Nested)
23   &MACRO EVENT FOCUS_IN 1 0, 1900544, 0, 0, 0! Task Requirements
24
25   #<EVT MUSCLK0055 0073511055646363>#
26   &MACRO EVENT FOCUS_IN 1 0, 3145731, 0, 0, 0! Breadth
27   &MACRO [Event ID] 31457 [Time]  RN 0) = "20"  ! Breadth
28
29   #<EVT MUSCLK0056 0073511055649843>#
30   &MACRO MODIFY_VERIFY_EVENT 1 0 3145731 1 STRN 0 2 insert = 4 proposed = 4 current = 20  ! Breadth
31   &MACRO EVENT VALUE_CHANGED 1 0, 3145731, 0, 0, 0! Breadth
32   &MACRO ASK_ITEM 3145731 (1 STRN 0) = "4"  ! Breadth
33   &MACRO MODIFY_VERIFY_EVENT 1 0 3145731 1 STRN 1 1 insert = 0 proposed = 40 current = 4  ! Breadth
34
35   #<EVT MUSCLK0057 0073511055654492>#
36   &MACRO MODIFY_VERIFY_EVENT 1 0 3145731 1 STRN 1 1 insert = 0 proposed = 40 current = 4  ! Breadth
37   &MACRO EVENT VALUE_CHANGED 1 0, 3145731, 0, 0, 0! Breadth
38   &MACRO ASK_ITEM 3145731 (1 STRN 0) = "40"  ! Breadth
39   &MACRO EVENT FOCUS_OUT 1 0, 3145731, 0, 0, 0! Breadth
```

Fig. 10. A section of the CAD log file is shown, with the event ID corresponding to a mouse click event and time.
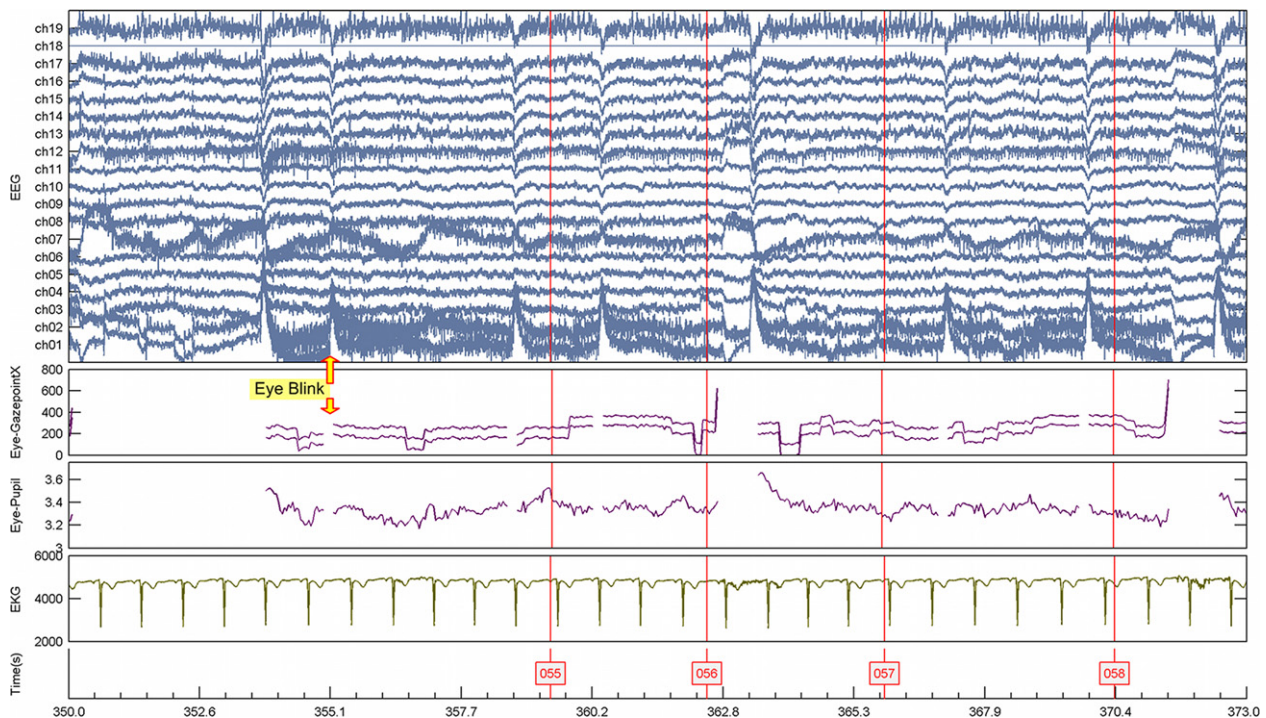

Fig. 11. Events generated by the CAD interface have been synchronously inserted in the EEG, EKG streams captured by a bio-physical monitoring device and Gaze point (left, right eyes) and pupil radius captured by an eye tracker. Events triggered by CAD software are indicated with unique event IDs. It can be visually examined that the eye blink—a physical action—captured by two independent devices is temporally aligned accurately.
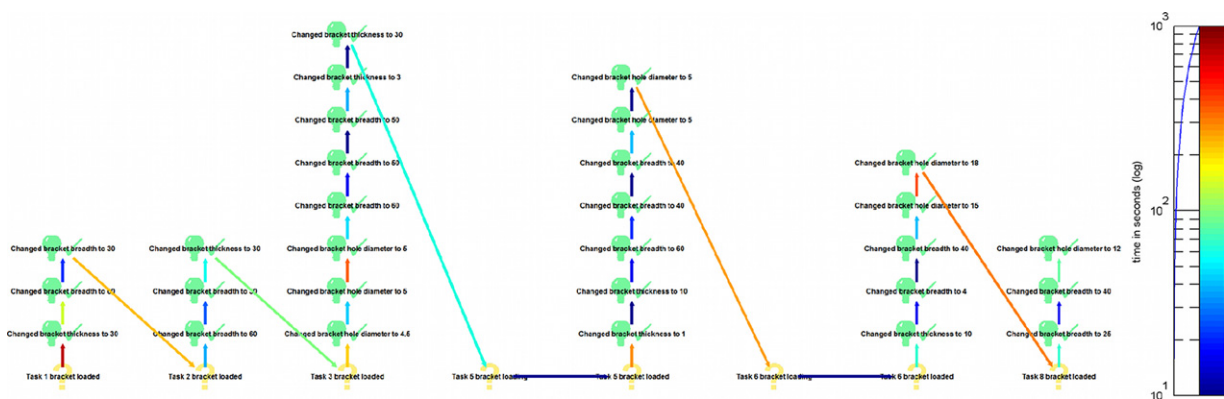

Fig. 12. A DRed Diagram, automatically generated by post processing the synchronised log file, showing 8 activities performed modifying the bracket design in the CAD task. Accurate time duration of each task has been illustrated by the colour code. The logarithmic scale has been used to point up both short and prolonged tasks. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
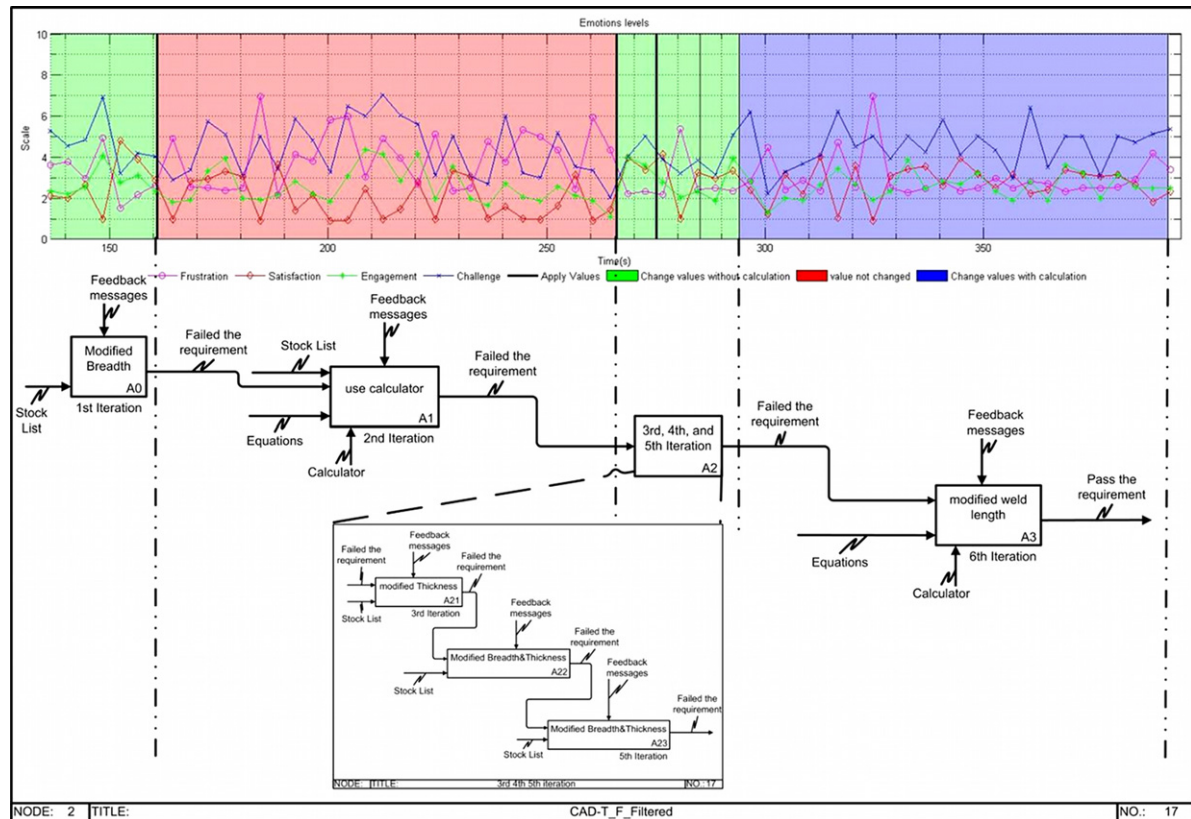
Fig. 13. IDEF0 diagram describing the iterations of participant while working on the Bracket Design Task, in conjunction with participant's emotional levels of frustration, satisfaction, challenge and engagement.

the user [36–40]. Employing the results from these studies in a ubiquitous CAD environment with multimodal signals is far from trivial. There are many technical challenges to overcome before building up such a complicated system and therefore this study centred on the technical resolution of multimodal ubiquitous data capture. Therefore the finer details of the psychophysiological data processing fall outside the scope of the paper.

Synchronisation-related evidences were presented which exhibit the time and chronological event synchronisation aspects of the framework. IDEF0 and DRed illustrations were provided to demonstrate how the metadata can be mapped onto the CAD task progress. These illustrations associated various stages and iterations of the design solution with the user state interpreted from EEG. In summary, the evaluation of the framework via this CAD task demonstrates the technical capability of the framework and its ability to capture metadata in an engineering design task. Further it has been shown that a standard CAD system can be extended to support sophisticated metadata capture, an application for which it was not originally designed

4. Use case 2: multimodal data capture in a collaborative design review

The capture of knowledge relating to a product throughout its lifecycle has been an important issue in engineering industry for many decades, especially in relation to formal design reviews. Traditional knowledge capture methods have typically relied on manual techniques that are time-consuming and disruptive to engineers, resulting in costly overheads [41,42]. Therefore, there has been widespread research into automating the process of capturing engineering knowledge and rationale, and previous work by the authors have successfully demonstrated this through user-logging in virtual design environments [9,13].

4.1. Virtual aided design engineering review system with metadata

The Virtual Aided Design Engineering Review (VADER) system addresses the shortcomings of current design review methods by the unobtrusive time-phased capture of multimodal data during individual and group activities performed in a design review. The VADER system developed by the authors allows the 3D visualisation and annotation of computer-aided design models, where all captured data can easily be viewed and searched post-review. Furthermore, review reports and formalised representations of the captured knowledge can be automatically generated and stored in traditional product lifecycle management systems for future reuse. The ubiquitous data capture framework serves as a basis to integrate, capture and synchronise multimodal inputs during the design review. Various multimodal inputs, i.e. text input, audio, video, 3D selection/highlight from centralised and remote/distributed locations are integrated into the metadata system. Fig. 14 illustrates the organisation of the capture devices integrated within the VADER system. The virtual reality interface for a design review room is shown in Fig. 15, whereas distributed participants can connect to the system using other simplified and web browser-based interfaces.

4.2. Metadata timeline

The captured metadata from multiple sources is stored along with the relevant part (or assembly) name. The text based annotations are stored in XML nodes where a link to CAD data also maintained; therefore the stored data can be accessed bi-directionally. The framework preserves the chronological gathering of captured annotations, events and data streams and populates a time line based on a Unix time source provided by the framework (Fig. 16).
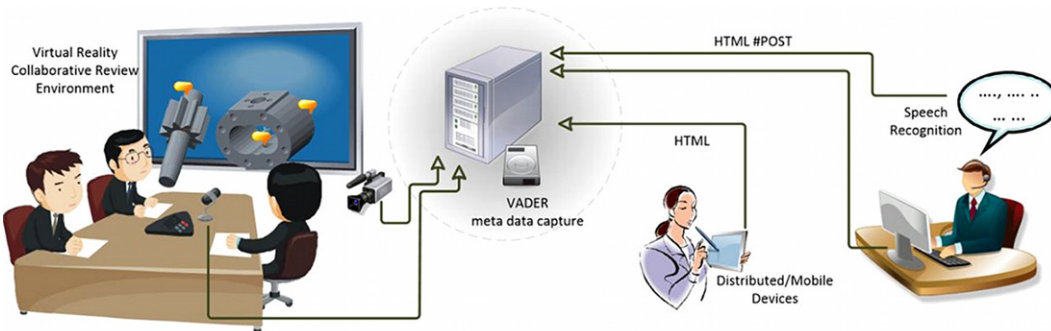
Fig. 14. VADER ubiquitous, multimodal metadata capture system. Various multimodal inputs from distributed locations are illustrated.
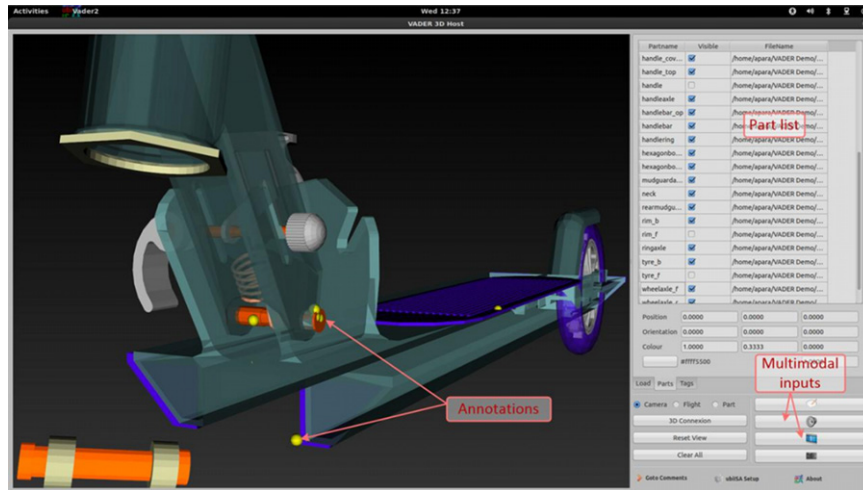

Fig. 15. VADER system's CAD visualisation interface. The framework allows multimodal annotations to be linked to the CAD parts with the temporal data. This information can also be reviewed in the metadata timeline.

The current system captures audio, video, keyboard and mouse input, but the modular nature of the capture framework permits extra input modalities to be easily added. The ubiquitous nature of the framework allows input devices to be plugged in and removed from the data capture system in runtime. This forms a perpetual metadata collection environment which runs in an open-ended fashion, where multiple people with various data interactive tools in distributed locations join and leave the environment as required.

4.3. Emphasis of the study

This use of the framework in a design review use case is intended to exhibit further capabilities of the framework in addition to the previous evaluation on a CAD system. The applicability of the framework in this environment proves its capability to serve the requirements of a multiuser collaborative environment. The application also demonstrated that users can be seamlessly connected from multiple, distributed sites simultaneously.

5. Discussion

The development of a CAD design is a complicated activity where designers gather information from various sources, perform computations and add value to the product by inputting their knowledge. Designers need to be supplied with sufficient information and rationale about the earlier designs and decisions so that they can make well informed decisions during product development. The tacit knowledge used by designers is valuable content which is also incorporated and implicit in the product. Capturing, transporting and formalising tacit knowledge across

time (i.e. from previous designs into latest designs) or places (e.g. from one person/team to another, different stages of designs, etc.) is not a straightforward task. Synchronously capturing various inputs, interaction and, biophysical data along with the formal design data is demonstrated in this paper with the view to facilitating the capture and transportation of performance data and knowledge.

The meta-data captured regarding the design activity provides new-levels of traceability of design knowledge, enabling better design decision making and design/design-rationale reuse. For instance a new engineer can discover not only what features were defined in a previous design but trace back why earlier engineers designed those features in a specific way and interrogate the engineers' discussions, interactions and cognitive processes during earlier designs. Not limited to capturing the data, this paper also demonstrates how the captured temporal meta-data can be represented in various formats, including IDEF, DRed and searchable multimodal timelines as exemplars to name but a few. Although structuring and organising the captured information into various representations is important, the thinking process of the designer or the design ideation does not follow a specific structure or sequence [43]. As a consequence, the holistic activity capture enabled by the multimodal framework and inherent process mapping provides new opportunities for exploring various methods for structuring and organising information which subsequently, after interpretation and understanding, becomes knowledge.

Conventionally monitoring CAD user activity is mainly based on the log files created by the CAD software. The availability of contemporary, sophisticated computer-based tools allow the accurate time-phase capture of user activity and provide opportunities
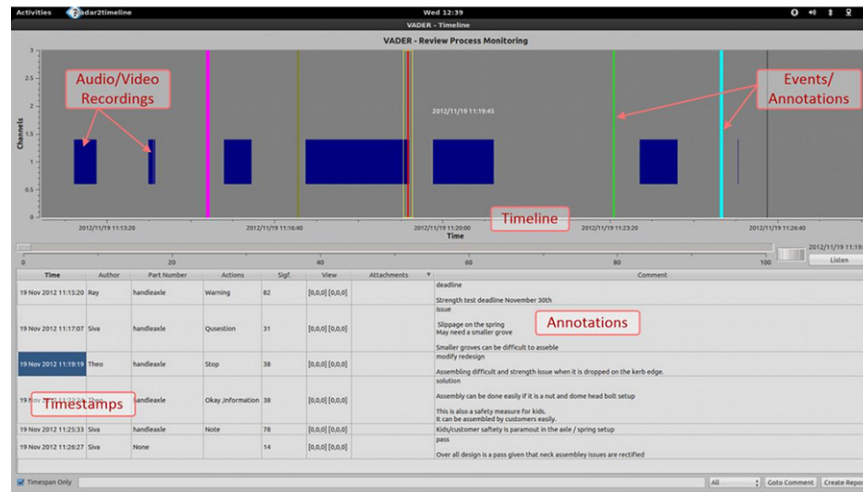
Fig. 16. Metadata timeline provided by the VADER system. This perpetual timeline can be revisited and investigated using search tools by engineers anytime in the products' lifecycle. The ubiquitous framework performs the core function of integrating and capturing multimodal inputs with the temporal information.

for understanding CAD design activities to unprecedented levels of detail. Multimodal ubiquitous data captured through a variety of devices, including CAD software logging, should be merged to generate meaningful data about the CAD activity. In contrast to a study of constrained user activity, CAD and the PLM (Product Lifecycle Management) are freeform and pro-longed processes, consequently forming the need for a ubiquitous metadata capture.

5.1. Ubiquitous metadata capture environment

Different representations of the experimental CAD activity and bio-physical data synchronised with the events occurring in the design environment demonstrate that capturing the design activity using ubiquitous multimodal media can provide a better insight into the actual design process. Similarly it has been demonstrated that building a collaborative engineering design review system with multimodal inputs and users from distributed sites is also possible. The ubiquitous framework presented in this paper enables capturing multimodal metadata, unobtrusively, i.e. without inhibiting the process by easily embedding various capture tools in the environment.

Additional concerns are also addressed by the ubiquitous framework when deploying in a realistic, unconstrained environment, e.g. a design office with many engineers. Since the design process is virtually a cyclical and long process, the data capture also needs to run continuously over a long period of time. The distinctive modular-based design of the framework enables it to run indefinitely. A base station or a centralised control panel is required to run uninterruptedly while individual device units can be started and stopped as required. Since the framework allows the plugging or removal of devices at runtime and caters for device failure, this makes the framework robust and suitable for a rugged environment such as a design office.

5.2. Interaction and synchronisation

In addition to capturing data, units that interact with the user or the environment can also be linked to the framework; in fact the framework would treat them indistinguishably from capture tools. Given that a unified interface has been provided for an interaction unit and uses the data flow modules to access data from other devices, it blends into the framework seamlessly. This feature potentially provides opportunities for knowledge push during a CAD task. Previously captured activity and the knowledge

committed to a knowledge base can be used to offer information to a designer automatically as and when the designer needs it. A real-time system actively monitoring for user activities can identify patterns of activity and retrieve information from similar previous activities captured and formalised in the past.

This study has proven that the framework synchronises multimodal sources and evidences were provided to demonstrate the achievable accuracy of synchronisation. In particular, when considering the technical requirements for a prospective brain computer interface (BCI) system for CAD, accurate synchronisation is critical between the operating system, the CAD application, the BCI device (EEG) and other modalities. This is fundamental because a BCI CAD system must be responsive but yet be able to tolerate various device protocols. The evaluation on the CAD system showed that the proposed framework is capable of handling this requirement through capturing and mapping EEG for design activities. The present study uses medical grade EEG and other psychophysiological sensors for safety, data provenance and hygiene reasons. As we understand more about design activities using a virtual environment (i.e. CAD) so too will development of collaborative design, BCI controls and indeed the functionalities in the virtual environments can be improved. Therefore a ubiquitous framework such as this addressing the fundamental technological challenges becomes very useful.

5.3. Handling vast quantities of metadata

Appropriate real-time fusion and processing modules can be incorporated in the system to produce ready-to-use data, resulting a reduction in the stored information compared to the raw data. For instance, a module for a video camera capturing the user activity only needs to store the recording when there is a presence of the user. Therefore plugging such a dynamic component into the system can possibly adapt the tools capturing the data, e.g. increasing or decreasing the data resolution. Early filtering of the captured content prevents inappropriate information being accumulated into the repository.

Currently the metadata, i.e. annotations and events is stored locally in individual files. Although this technique caters for the high bandwidth and low latencies, it causes difficulties when the captured data is accessed in real-time or shared with multiple users. As a solution authors consider methods of storing the captured data in centralised repository databases in future studies. The balance between distributed local files and centralised repositories is

needed to be maintained in order to take the benefits of both techniques. Furthermore, organisational structures and bureaucracies might require control over what is committed into the repository and moderation steps on the captured metadata before it is committed into mainstream repositories. Therefore, the authors recommend adding additional features to the framework in the future to handle the data accessibility.

5.4. Integrating metadata capture in existing and future CAD solutions

Present-day CAD solutions are typically not equipped for multimodal metadata inputs. A metadata capture system cannot be limited to ad-hoc configurations; instead it needs to be limitless with regard to the number and types capture devices. An ideal unbounded system should be open to interfacing with various multimodal tools and support different configurations of the metadata capture system. Therefore a future CAD system with metadata capture must adopt standard protocols such as the ubiquitous framework proposed in this paper. This will permit future CAD systems not to be selective on tools but to accommodate diverse multimodal tools to capture activities and interact with the design process. Consequently the users can select their own capture tools and mount those onto the system to construct a customised configuration of a metadata capture system.

Although PLM systems provide few facilities to handle multimodal data e.g. audio, video, hyperlinks to other data types, etc., they are obtrusive as a user needs to upload the information collected during the design. This interrupts the flow of the design process and somebody has to do the additional work of capturing and gathering the metadata. In contrast, the ubiquitous framework outfits the design environment with embedded data capture tools and hence provides a metadata collection system which is unobtrusive to the process. Two possible configurations are proposed for using the framework for metadata capture in a design environment.

1. *Embedding the metadata capture tools within the existing CAD systems*

The framework can be used to extend the existing CAD solutions to include ubiquitous metadata capture, whereas multimodal inputs will be captured alongside the CAD design trail (Fig. 17). This is similar to the case of metadata capture using a CAD station as demonstrated in this paper.

2. *Building a metadata capture infrastructure based on the ubiquitous framework.*

A complete metadata capture infrastructure can be constructed using the framework. This will comprise ubiquitous data capture tools and standard CAD solutions. In this arrangement any CAD solution will be considered as an element of the whole system (Fig. 18). This is similar to the case of design review system outlined in this paper.

In both cases the metadata and updating PLM relations are driven automatically by the framework. Embedding the metadata capture tools within the standard CAD systems is a quicker and a simpler way to implement a metadata system but it results in limitations over the extendibility. This is more suited for a single CAD station (single user) but limited to the customisation features such as SDK or API provided by a proprietary CAD solution e.g. PTC Creo TOOLKIT, Siemens NX/UG Open API. In contrast, constructing a complete metadata infrastructure is more suited for multiuser, parallel sessions, and distributed sites metadata collection system. Thus, it comes with the cost of implementing a complete ubiquitous capture infrastructure and adopting the information generated, i.e. raw metadata such as audio files, video files, and key press logging or processed data such as filtered activities and emotional states into the current PLM systems.

5.5. Feeling of being watched

Capturing metadata during various stages of the design activity provides unprecedented insights about the design process and traceability of the design solution. However capturing user activities can potentially have adverse effects on the employees (i.e. designers and engineers) such as being suspicious about the employer and possibly causing a breakdown of trust. This topic has been discussed in several studies with a view to addressing the shift in technologies, workplace privacy and ethics [44–47]. It is rather common to have sessions or discussions "off the record" from minutes in various stages of a product design. Similarly in the collaborative design review task presented (Section 3.5) engineers wanted to discuss things during the review session which they did not want to be included in the metadata collection system. The framework allows capture devices to be enabled and disabled by the user in runtime, i.e. being dynamic and reconfigurable as opposed to being a fixed system. This makes it possible for an engineer to walk into the ubiquitous environment and start (or stop) capturing his/her own work. Therefore it imitates an atmosphere where an individual can sign-in and sign-out allowing individuals to have control over the capture of his/her own activities. It is also possible that the captured metadata might be used to evaluate the employer performance. However this issue is associated with the organisational and social aspects of how this information is used, whether for revisiting and tracing of the design knowledge and product information or for evaluating employee performance.

6. Conclusion

This paper has presented novel ways of capturing the metadata about the design process beyond the standard design records such as CAD parts, geometries, and manually produced reports and minutes. The metadata is captured and stored using ubiquitous multimodal capture tools embedded in the design environment. This data capture is automatic and blended into the habitual design activity and therefore unobtrusive to the process. Consequently, the designer/engineer has no interruption in the usual design activity and particularly, there is no extra workload involved in capturing and generating the meta-information about the design process.

The generic ubiquitous framework presented in this paper establishes the metadata capture system and drives the automatic and ubiquitous capture of temporal multimodal data. The framework forms a highly expandable system with systematic arrangement of data capture tools. It provides a standardised, repeatable method for integrating multimodal, ubiquitous devices and hence enabling the progressive construction of a complicated data capture infrastructure. The approach maintains a generic architecture for multimodal data capture and enables the construction of systems for metadata in diverse environments with disparate obligations. In contrast to related studies, e.g. DAKA [4] and KEN [2] which deal with the captured metadata, this work addressed a detailed technical resolution for capturing multimodal metadata and demonstrates an end-to-end pipeline from the capture of raw measurements data to metadata representations mapped onto the design process, all accessible and generated automatically.

Nevertheless, the ubiquitous framework itself is not restricted to CAD. Its generic approach means that it can be applied into other scenarios such as psychological and user behaviour monitoring systems, game activity monitoring, driving simulation, virtual reality environments, etc. Since the core of the framework is an abstract definition, various application configurations can be built upon the framework using it as a template. Modules in the framework have been designed to be extended to support new
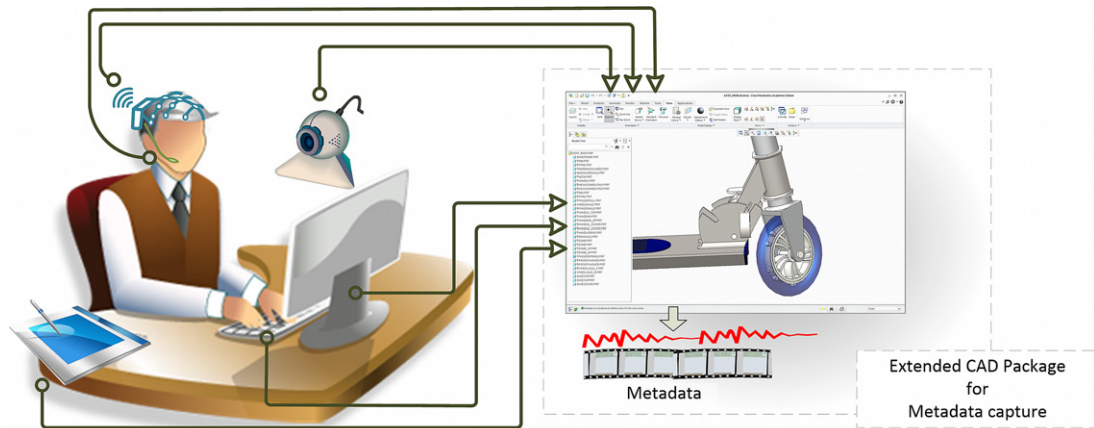
Fig. 17. Extending CAD station for multimodal metadata capture. CAD package can be customised using the framework to accommodate multimodal tools.
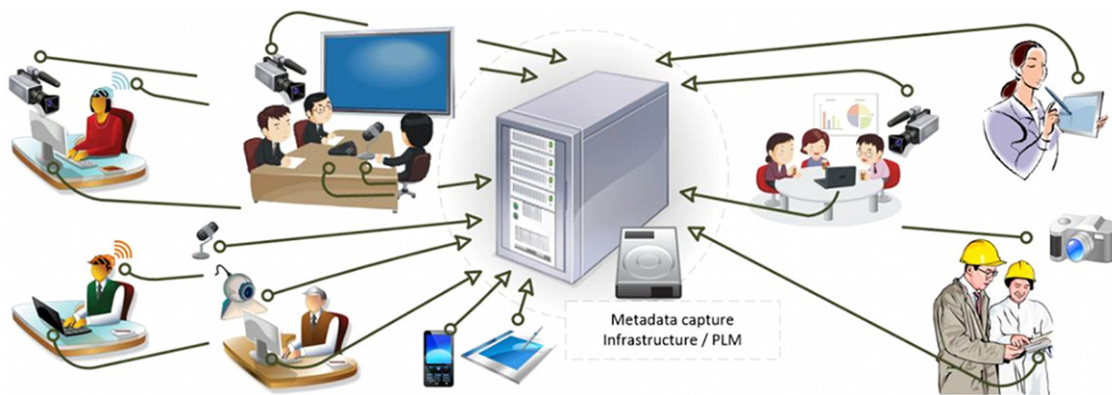

Fig. 18. Complete metadata capture infrastructure with ubiquitous devices placed in the design environment and also in remote sites.

devices and data types, effectively allowing any programmer to build a temporal data capture system according to their associated scenario requirements. The framework principally solves technical issues related to constructing a ubiquitous data capture system so that individual researchers and programmers do not have to reinvent the wheel over and over again.

Acknowledgements

References

[1] Bechara A, Damasio H, Damasio AR. Emotion, decision making and the orbitofrontal cortex. Cerebral Cortex 2000;10:295–307.
[2] Conway AP, Ion WJ. Enhancing the design dialogue: an architecture to document engineering design activities. Journal of Engineering Design 2013; 24:140–64.
[3] Szykman S, Sriram RD, Bochenek C, Racz JW, Senfaute J. Design repositories: next-generation engineering design databases. 2000.
[4] Jin Y, Ishino Y. DAKA: design activity knowledge acquisition through data-mining. International Journal of Production Research 2006;44:2813–37.
[5] Shipman FM, McCall RJ. Integrating different perspectives on design rationale: supporting the emergence of design rationale from design communication. Artificial Intelligence for Engineering, Design, Analysis and Manufacturing 1997;11:141–54.
[6] Campbell DR, Culley SJ, McMahon CA, Sellini F. An approach for the capture of context-dependent document relationships extracted from Bayesian analysis of users' interactions with information. Information Retrieval 2006;10: 115–41.
[7] Campbell DR, Culley SJ, McMahon CA, Coleman P. A methodology for profiling computer based design activities. In: ICED 05: 15th international conference on engineering design: engineering design and the global economy 318. 2005.
[8] Huet G. Design transaction monitoring?: understanding design reviews for extended knowledge capture (Ph.D.). University of Bath. 2006.
[9] Conway A, Wodehouse A, Ion W, Juster N. A study of information & knowledge generated during engineering design meetings. In: international conference on engineering design. Presented at the international conference on engineering design. 2007.
[10] Rea HJ, Howley IK, Corney JR, Ritchie JM, Sung R, Salamon C. CBBC BAMZOOKi™ as a tool for engineering design research. In: Proceedings of learning with games (LG2007), Sophia Antipolis, France. 2007. p. 585–94.
[11] IDEF0. Wikipedia, the free encyclopedia. 2013.
[12] Bracewell RH, Ahmed S, Wallace KM. DRed and design folders: a way of capturing, storing and passing on — knowledge generated during design projects. In: Proceedings of the 2004 ASME international design engineering technical conferences (DETC'04). Vol. 1. Presented at the 2004 ASME international design engineering technical conferences (DETC'04), ASME: American Society of Mechanical Engineers, Salt Lake City, Utah, USA, 2004. p. 235–46.
[13] Ritchie JM, Sung R, Rea H, Lim T, Corney JR, Salamon C. et al. Automated knowledge capture in 2D and 3D design environments. In: 2nd International workshop virtual manufacturing VirMan 08 as part of the 5th INTUITION international conference: virtual reality in industry and society: from research to application. 2008.
[14] Sung RCW, Ritchie JM, Lim T, Kosmadoudi Z. Automated generation of engineering rationale, knowledge and intent representations during the product life cycle. Virtual Reality 2011;16:69–85.
[15] Atterer R, Wnuk M, Schmidt A. Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In: Proceedings of the 15th international conference on world wide web. 2006. p. 203–12.
[16] Dragunov AN, Dietterich TG, Johnsrude K, McLaughlin M, Li L, Herlocker JL. TaskTracer: a desktop environment to support multi-tasking knowledge workers. In: Proceedings of the 10th international conference on intelligent user interfaces. 2005. p. 75–82.
[17] Stumpf S, Bao X, Dragunov A, Dietterich TG, Herlocker J, Johnsrude K. et al. Predicting user tasks: I know what you're doing. In: 20th National conference on artificial intelligence (AAAI-05), workshop on human comprehensible machine learning. 2005.

[18] Hilbert DM, Redmiles DF. Extracting usability information from user interface events. ACM Computing Surveys (CSUR) 2000;32:384–421.

[19] Fenstermacher KD, Ginsburg M. A lightweight framework for cross-application user monitoring. Computer 2002;35:51–9.

[20] Kaptelinin V. UMEA: translating interaction histories into project contexts. In: Proceedings of the SIGCHI conference on human factors in computing systems. 2003. p. 353–60.

[21] Lokaiczyk R, Faatz A, Beckhaus A, Goertz M. Enhancing just-in-time E-learning through machine learning on desktop context sensors. In: Kokinov B, Richardson D, Roth-Berghofer T, Vieu L, editors. Modeling and using context. Lecture notes in computer science, Berlin (Heidelberg): Springer; 2007. p. 330–41.

[22] Hercegfi K, Kiss O, Bali K, Izso L. Interface: assessment of human–computer interaction by monitoring physiological and other data with a time-resolution of only a few seconds. In: ECIS 2006 Proceedings. 2006.

[23] Bierig R, Gwizdka J, Cole MJ. A usercentered experiment and logging framework for interactive information retrieval. In: Proceedings of the SIGIR 2009 workshop on understanding the user: logging and interpreting user interactions in information search and retrieval. 2009. p. 8–11.

[24] Heger D, Putze F, Schultz T. Online workload recognition from EEG data during cognitive tests and human–machine interaction. In: Dillmann R, Beyerer J, Hanebeck U, Schultz T, editors. KI 2010: advances in artificial intelligence. Lecture notes in computer science, Berlin (Heidelberg): Springer; 2010. p. 410–7.

[25] Hercegfi K, Pászti M, Tóvölgyi S, Izsó L. Usability evaluation by monitoring physiological and other data simultaneously with a time-resolution of only a few seconds. In: Human–computer interaction. New trends: 13th international conference, HCI international 2009, proceedings. Berlin (Heidelberg): Springer-Verlag; 2009. p. 59–68.

[26] Müller KR, Tangermann M, Dornhege G, Krauledat M, Curio G, Blankertz B. Machine learning for real-time single-trial EEG-analysis: from brain-computer interfacing to mental state monitoring. Journal of Neuroscience Methods 2008;167:82–90.

[27] Rezazadeh IM, Wang X, Firoozabadi M, Hashemi Golpayegani MR. Using affective human–machine interface to increase the operation performance in virtual construction crane training system: a novel approach. Automation in Construction 2010.

[28] Buscher G, Gwizdka J, Teevan J, Belkin N, Bierig R, Elst L. et al. Preface: Proceedings of the SIGIR 2009 workshop on understanding the user — Logging and interpreting user interactions in information search and retrieval. Proceedings of the workshop on understanding the user — logging and interpreting user interactions in information search and retrieval (UIIR-2009) i–v. 2009.

[29] Akahoshi Y, Kidawara Y, Tanaka K. A database-oriented wrapper for ubiquitous data acquisition/access environments. In: Proceedings of the 2nd international conference on ubiquitous information management and communication. 2008. p. 25–32.

[30] Goudeseune C, Kowitz B. Synchronous data collection from diverse hardware. In: Conférence Simulation de Conduite. 2004. p. 245–52.

[31] Soleymani M, Lichtenauer J, Pun T, Pantic M. A multi-modal affective database for affect recognition and implicit tagging. IEEE Transactions on Affective Computing 2011.

[32] Dubey A, Karsai G, Abdelwahed S. Compensating for timing jitter in computing systems with general-purpose operating systems. IEEE; 2009. p. 55–62.

[33] Plant RR, Hammond N, Whitehouse T. How choice of mouse may affect response timing in psychological studies. Behavior Research Methods 2003; 35:276–84.

[34] Sung R, Ritchie JM, Rea HJ, Corney J. Automated design knowledge capture and representation in single-user CAD environments. Journal of Engineering Design 2011;22:487–503.

[35] Liu Y, Kosmadoudi Z, Sung R, Lim T, Louchart S, Ritchie J. Capture user emotions during computer-aided design. In: Proceedings of IDMME - virtual concept 2010. Presented at the IDMME - Virtual Concept, 2010.

[36] Wang X-W, Nie D, Lu B-L. EEG-based emotion recognition using frequency domain features and support vector machines. In: Lu B-L, Zhang L, Kwok J, editors. Neural information processing. Lecture notes in computer science, Berlin (Heidelberg): Springer; 2011. p. 734–43.

[37] Van Nimwegen C, Uyttendaele AJ. Unobtrusive physiological measures to adapt system behavior: the GSR mouse. In: status: published. presented at the key issues in sensory augmentation, Institute for Media Studies, KU Leuven Nederlands, Brighton, Sussex, UK. 2009.

[38] Partala T, Surakka V. Pupil size variation as an indication of affective processing. International Journal of Human-Computer Studies 2003;59: 185–98.

[39] Ayzenberg Y, Hernandez Rivera J, Picard R. FEEL: frequent EDA and event logging–a mobile social interaction stress monitoring system. In: Proceedings of the 2012 ACM annual conference extended abstracts on human factors in computing systems extended abstracts. 2012. p. 2357–62.

[40] Klein J, Moon Y, Picard RW. This computer responds to user frustration: theory, design, and results. Interacting with Computers 2002;14:119–40.

[41] Ameri F, Dutta D. Product lifecycle management: closing the knowledge loops. Computer-Aided Design & Applications 2005;2:577–90.

[42] Bernard A, Laroche F, da Cunha C. Models and methods for knowledge formalisation in a PLM context. In: Actes de La Conférence CMSM'2009. 2009.

[43] Chandrasegaran SK, Ramani K, Sriram RD, Horváth I, Bernard A, Harik RF, et al. The evolution, challenges, and future of knowledge representation in product design systems. Computer-Aided Design 2013;45:204–28.

[44] Langheinrich M. Privacy in ubiquitous computing. Chapman and Hall/CRC; 2013.

[45] Hong JI, Landay JA. An architecture for privacy-sensitive ubiquitous computing. In: Proceedings of the 2nd international conference on mobile systems, applications, and services. 2004. p. 177–89.

[46] Nord GD, McCubbins TF, Nord JH. E-monitoring in the workplace: privacy, legislation, and surveillance software. Communications of the ACM 2006;49: 72–7.

[47] Miller S, Weckert J. Privacy, the workplace and the internet. Journal of Business Ethics 2000;28:255–65.