The Collective Experience of Empathic Data Systems

ICT-258749

Deliverable 3.1

CEEDs engine: definitions, architecture, narratives and data discovery

---

Authors	Pedro Omedas, Ulysses Bernardet, Paul Verschure, Michalis Makridis, Theodoros Semertzidis, Petros Daras, Frederic Fol Leymarie, Eckehard Olbrich, David Pizzi
Version	f1.0
Date	30.09.2011
Classification	Restricted
Contract Start Date	01.09.2010
Duration	48 months
Project Co-ordinator	Goldsmiths, University of London
File Name	<CEEDs_D3.1_f1.0_UPF>

Project funded by the
European Community under
the Seventh Framework Programme

Consisting of:

No	PARTICIPANT NAME	S.N.	COUNTRY
1	Goldsmiths, University of London	GOLD	UK
2	Universitat Pompeu Fabra	UPF	ES
3	University of Sussex	UOS	UK
4	Informatics and Telematics Institute	ITI	GR
5	Eberhard Karls Universitaet Tuebingen	EKUT	DE
6	Universität Augsburg	UAU	DE
7	University of Teesside	TEESSIDE	UK
8	Università degli Studi di Padova	UNIPD	IT
9	Max Planck Gesellschaft zur Foerderung der Wissenschaften E.V.	MPG	DE
10	Ecole Normale Superieure	ENS Paris	FR
11	Budapesti Muszaki Es Gazdasagtudomanyi Egyetem	BME	HU
12	Universitat Politecnica de Catalunya	UPC	ES
13	Università di Pisa	UDP	IT
15	Electrolux Italia SpA	ELECTROLUX	IT
16	Leiden University	UL	NL
18	Helsingin Yliopisto	UH	FI

Responsible of the document: UPF

Defined Contributors to the document: GOLD, UPF, ITI/CERTH, TEESIDE, UNIPD, MPG, UPC

Document History

VERSION	ISSUE DATE	BY	CONTENT AND CHANGES
V0.1 Draft	18/08/2011	UPF	First draft of the document: overall content and CEEDs Engine architecture (UPF), Data Discovery (MPG)
V0.2 Draft	30/08/2011	CERTH	Added contribution to: "Archaeology application", "application domains and type of data"
V0.2 Draft	19/09/2011	GOLD	Added contribution to: "Archaeology application"
V0.3 Draft	19/09/2011	UPF	Integrated contribution from CERTH and GOLD. Added content for CSA.
V0.4 Draft	22/09/2011	TEESIDE	Added description for Narrative generator
V0.5 Draft	29/09/2011	UPF	General revision and correction of figure numbers and tables
V0.6 Draft	17/10/2011	UPF	General revision
f1.0 Draft	17/10/2011	UPF	Final document

Executive Summary

The objective of this work package is to design, build and test the CEEDs core engine. This engine orchestrates the interaction between the users of CEEDs and the data space that will be explored. Key elements of the engine are generic data mining, discovery and analysis tool, narrative structure generators, and the CEEDs Sentient Agent (CSA), an intentional, sentient autonomous agent.

This deliverable focuses on the conceptualization of the CEEDs engine architecture, the narrative generator and the data discovery. During the reporting period the conceptual architecture of the CEEDs engine, the narrative generator and data discovery components were defined. Additionally, the conceptualization work of other components such as the CEEDs Sentient Agent and the Composite Engine were started. The CEEDs engine has been developed working closely with other work packages in order to create a common framework that is a reference for the CEEDs applications (see D6.1). Through the progress of the project this architecture will be revised and adapted. To provide adequate methods and algorithms the data discovery component and the narrative generator were developed in close collaboration with the application developers. These two components are already being integrated in the actual application prototypes.

As part of the cooperation with other work packages a technical integration meeting was organized that served to bring together all technological partners to discuss the architecture of the engine and the applications. The meeting also served to start practical work on the development and integration of technologies.

Table of Contents

List of Figures

List of Tables

1 Introduction

CEEDs contributes to addressing the question of how we can advance our understanding of the world and the data we extract from it, by placing human experience at the centre of the solution. To address this question CEEDs develops and deploys an integrated theoretical and conceptual framework. CEEDs will provide novel ways for the on-line perception, creation of an interaction with massive volumes of data based on the enactive merging of physical and virtual sources of stimulation will deliver.

This document describes the CEEDs engine, the conceptual framework of the project that supports the realization of the theoretical framework developed in WP1. The architecture of the CEEDs engine serves as a common concept across all CEEDs applications developed WP6.

1.1 The process of human data exploration and discovery

We assume human data exploration and discovery can be described as a closed-loop system (Figure 1), from the original data to the visualization and storage of user reactions that influence the visualization of new data. In this process the Data is measured from the object of interest (cloud) and stored as raw data in a database. The user requests data via a query. From the database the dataset is filtered and presented to the user. Based on this presentation of the data the user performs cycles of hypothesis generation, that are based on induction, deduction, and abduction. Once satisfied with the hypothesis (illumination), the user verifies the hypothesis. For this step a new set of measurements might be required. If the result of the verification does not satisfy the desired level of understanding, the user will return to the development of a new hypothesis. In this schema abduction can be seen as the "injection" of hypotheses from outside of the user's own hypothesis generation. In the diagram thick lines denote connection of special relevance for the CEEDs system.

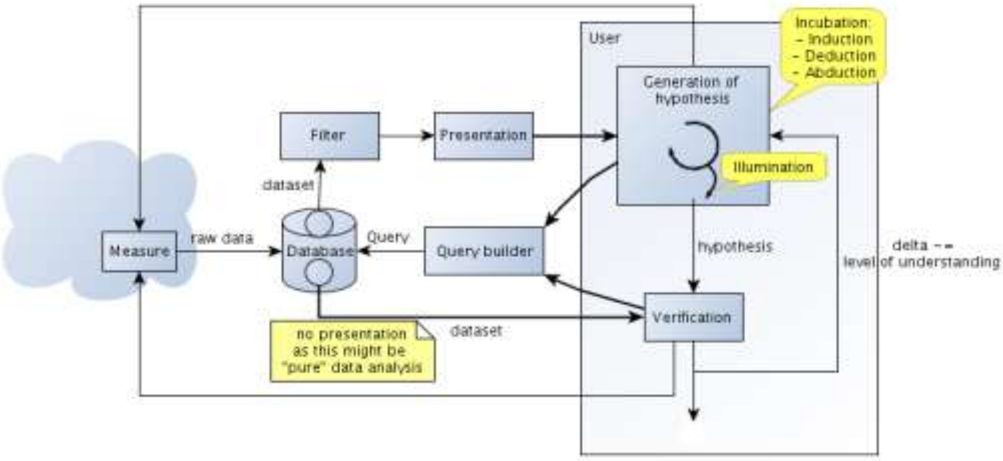


Fig. 1 - *Systematic view of human data exploration and discover.*

2 CEEDs Engine: Architecture and definitions

The CEEDs engine is the conceptual framework of the project that will provide a common architecture to CEEDs applications and is the result of task 3.1. The internal processing of the CEEDs engine is based on three main components. Firstly, adaptive data mining and discovery that defines what data is presented to the user. Secondly, spatio-temporal structuring of the presented content in the form of narratives generated by the composition engines, and thirdly, the "CEEDs Sentient Agent" (CSA), an intentional, sentient agent who controls the interface and can guide data exploration. A set of core features based on the use and implementation of this components have been defined by WP8 to clarify the architecture of the engine and the implementation of CEEDs applications, these core features are described in details in D8.1. For the importance of these definitions these core features are briefly described in the next section.

2.1 Core features

The CEEDs core features describe the requirements of an application to be considered a CEEDs application. The engine design supports the core features and provides a framework for their implementation. Throughout its development a CEEDs application can incrementally incorporate the core features. The following core features are defined:

2.1.1 [CF-RDDB] Raw Data Database

A database that stores the data available from the stakeholder. This data forms the basis of what can be explored by the user. The term "raw data" stems from the fact that it has not been processed by the CEEDs Engine. This does not exclude preprocessing for the data by elements outside the CEEDs context. The concrete content of the database depends on the application area.

2.1.2 [CF1] Display of CSA[1]-independent filtered view, perspective or flow of raw data.

An application implementing this core feature is able to display cue sequences of data stored in RDDB independent of the CSA agent. The display can be active or passive. Passive sequences are based on: (a) other variables e.g., sort, match, typology, reconstruction; or (b) directorial/producer preference, or (c) random. The active interaction is based on: subliminal or supraliminal influence/guide users' experience of the display.

---

[1] CEEDs Sentient Agent

2.1.3 [CF2] Recording of user implicit and/or explicit responses.

Users (a) explicit and/or (b) implicit responses when interacting with the CEEDs s system are recorded. Signals can be collected from a CSA independent [CF1] or dependent views [CF4].

2.1.4 [CF3] Interpretation and storage of the output of CF2

Datasets created from user implicit/explicit responses [CF2] are interpreted in a high level semantic way and stored in a database [CF-URDB].

2.1.5 [CF-URDB] User response database

The CEEDs application implements a database that stores outputs of users responses (CF2 and CF3), in relation to raw data (input to CF1).

2.1.6 [CF4] Display based on a user model of a CSA-dependent view, perspective or flow of a raw dataset

An application implementing this core feature displays cue sequences rule based dependent on CSA. The associated metadata influences presentation of raw data to the user. Subliminal or supraliminal influence/guide users experience of the display. Based on outputs of CF2, CF3 (based on outputs of CF1) on which sort, match, typology functions could be applied (empathy, understanding).

2.1.7 [CF5] Overlay CSA input data and user responses

Display user's responses and/or the data on which the CSA is making decisions as an overlay to the display data (CF1 or CF4). It is based on outputs of CF2, CF3 (overview, planning, professional, deliberate). It can also display metadata.

2.2 Role of the CEEDs system in the discovery process

The CEEDs system and the derived applications are the "CEEDs abduction" tool. The system fulfils this role by tapping into the discovery process (**Error! Reference source not found.**). This is aligned with the core features as follows:

[CF1] Support for the "normal" discovery process in that the engine allows the user to interact with raw data (CF-RDDB). Here the user interacts by means of explicit signals.

[CF4] Using implicit signals (CF2), that are semantically decoded (CF3), the CEEDs engine provides the user with a novel view of the raw data. These novel views serve to foster the creation of new hypotheses (abduction), and are based on models of the user, and records of user responses (CF_URDB). The content of the novel view is provided by data discovery tools.

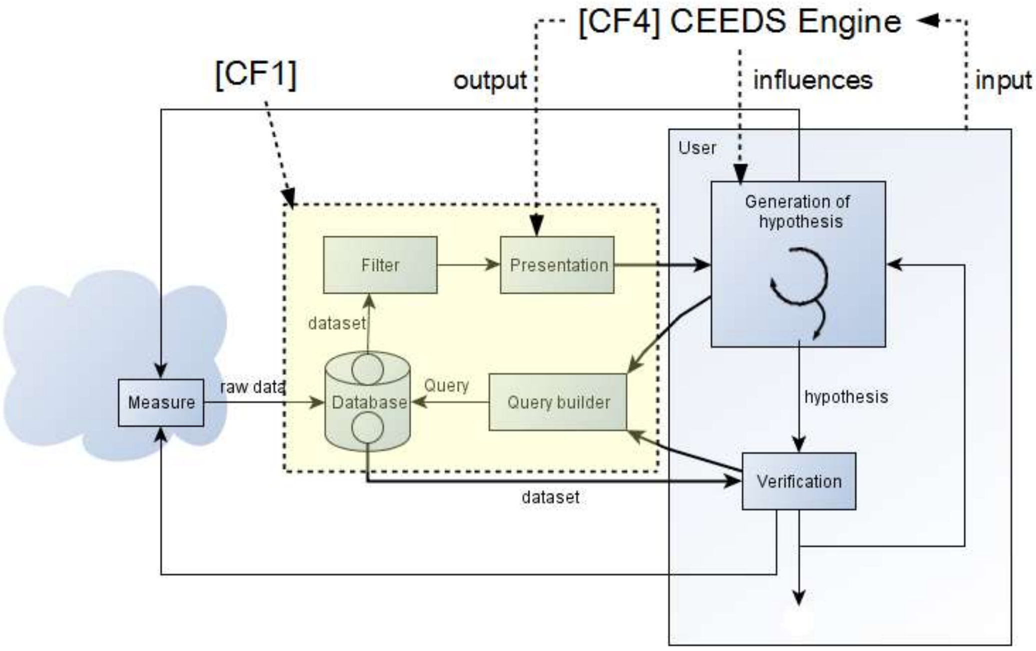


Fig. 2 - *Role of the CEEDs engine in the process of data exploration and discovery.*

2.3 Characteristics of CEEDs Engine

CEEDs-like systems can be classified along a number of characteristic. Some characteristics are mutually exclusive, while others can be combined.

System type
The system can either be open-loop, meaning it does not respond to input from the user, or closed-loop where the behaviour of the user influences the behaviour of the system.

Autonomy
The CEEDs system can either be non-autonomous or autonomous. Autonomy means that the system has a defined goal that it tries to achieve. Goal pursuit is a core characteristic of the CEEDs Sentient Agent (CSA).

Input signal types
The signal that is recorded from the user is either explicit or implicit. An explicit signal means that, from the user's perspective, the response of the system is an explicit reaction to the action of the user. An implicit signal means that from the user's perspective the behaviour of the system is under-determined, i.e. the user has no voluntary control over the signal he/she generates.

Signal origin

The user signal can either be stemming from the user that is currently interacting with the system ("own signal"), or from a user that previously interacted with the system ("other's signal"). The engine will provide a process to interface to devices that capture users' signals, and store it for their posterior use. The Engine will have a process to easily add, remove and upgrade new devices capturing the signals, developed in the project by WP2. Implicit signals used for the [CF1] architecture are 1) bio-signals including: electrophysiology M/EEG, Electro Dermal Activity (EDA), Electrocardiogram (ECG), etc. 2) behavioural cues (trajectory, position, posture, eye tracking, gaze, etc.).

The engine will receive also explicit inputs from users and will have support for different devices that will capture this instruction, such as button pressing, manipulation of tangibles, gestures, voice, etc.

Presentation types What is presented to the user can be perceived either subliminal or supraliminal. While supraliminal mean the user is fully aware of all aspect of what is presented, subliminal means that information is presented such that the user is not fully aware of all the information.

The relation of the characteristics with the core features CF1 and CF4/5 are show in **Error! Reference source not found.**. The presence of an autonomous agent that pursuits a goal (CSA) indicate if the application implements [CF1] or [CF4/5].

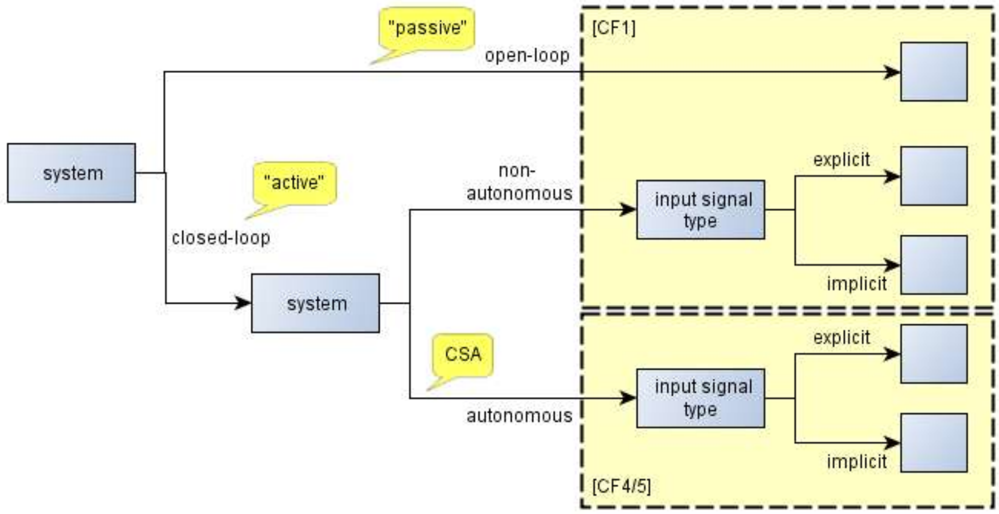


Fig. 3 - *Classification for different types of data presentation systems and their relationship to the CEEDs core features.*

The characteristics of the CEEDs system can be directly mapped onto the architecture of the CEEDs Engine (**Error! Reference source not found.**). The diagram shows how the user interacts in general with a CEEDs system; the CEEDs engine receives as input the data from the own user or other users and the engine provides a sequence of data to present to the user, influence by the goal of the autonomous system. The user experiences the presentation and produce explicit or implicit signals that are interpreted and feedbacks the engine to close the loop.

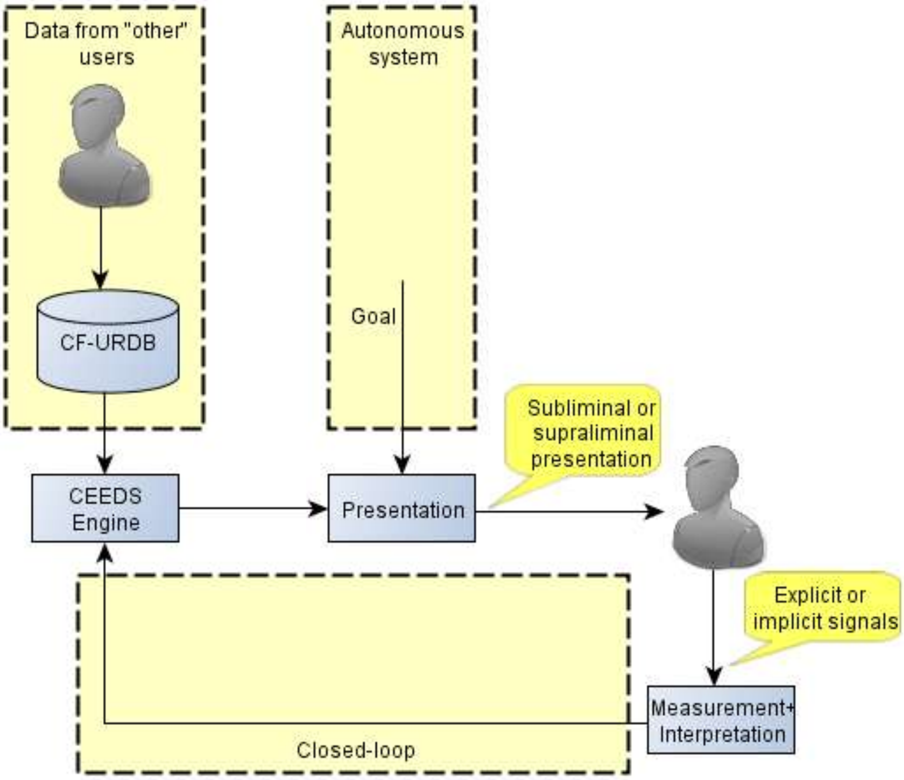


Fig. 4 - *Relationship between system characteristics and CEEDs system architectures. Yellow boxed show optional elements, and how these elements relate to the function of the system*

2.4 Architecture

The architecture of CEEDs engine is conceptually designed to support the development by building layers of functionality mapping to the DAC model. The lowest layer is the reactive and is a direct mapping between the user behaviour and the effectors. Next is the adaptive layer where the output of the engine is adapted to the past and presented behaviour of one or more user. The contextual layer provides novel interactions based on the integration of contextual information. Finally the reflective layer is at the top and implements the highest level of autonomy of the CEEDs engine by integrating the CSA.

The components of the engine are built in parallel to CEEDs applications so it will allow starting to work on the applications while the components are developed. The engine supports common components and communication protocols that each application will implement in their concrete scenario. In this way CEEDs applications will share a common architecture.

As a first stage the engine will support the development of 2.1.2 or the reactive layer, allowing displaying raw data to the user supporting explicit signals and symbolic commands from the user such as controllers, gesture, and speech commands. A database that stores the raw data [CF-RDDB] can be queried from the Filter or the Data Discovery module; it is assumed that the CF-RDDB contains "amorphous data", i.e. data that does not contain any higher order relationships.

The data can be presented to the user as requested i.e. no higher order structures are extracted from the data, or through the Narrative Generator and the Composition Engine that will extract patterns using a data mining component, and delivering the content with a narrative structure (the adaptive layer). Both cases need a filter/selection element, and need to yield visualizations. Additionally both cases should receive input from the user (Figure 5).

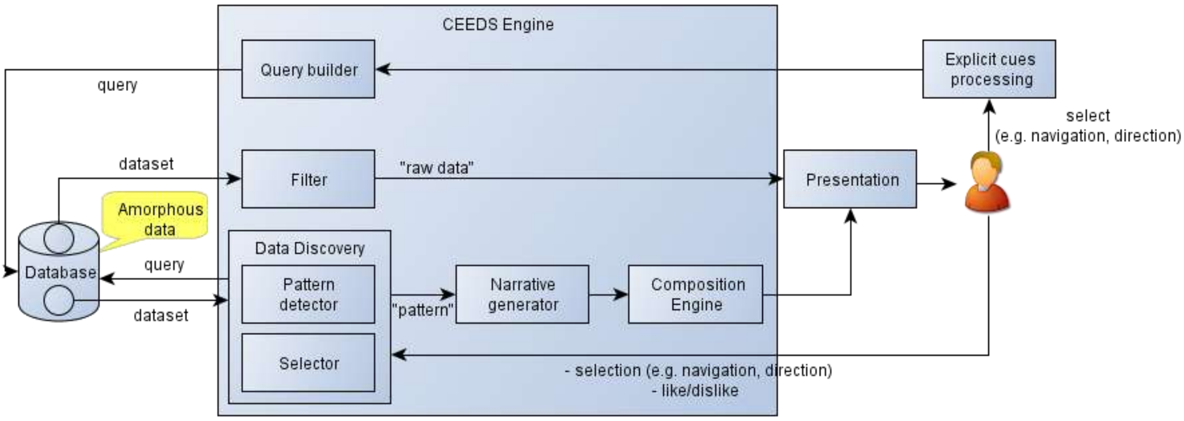


Fig. 5 - *Architecture of the reactive layer of the engine that is geared towards [CF1]*

In a second stage the system incorporates the components that will allow implementing CF4 and CF5, by including the CSA component to the architecture implementing the contextual and reflective layer. In addition to CF1 the engine will include a database (CF-URDB) to store user implicit and/or explicit responses and will be associated to the content that the user is reacting (delivered by the composition engine). The CSA will use this information and the user model to induct new hypothesis by sending commands to the Data Discovery module.

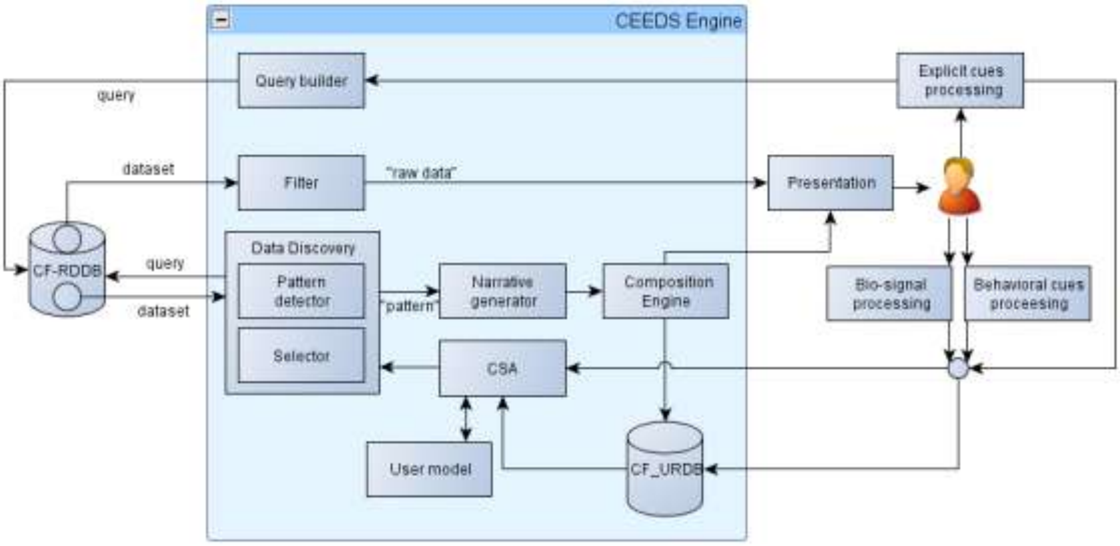


Fig. 6 - *Architecture of the engine to support [CF4]*

3 Data Discovery

Data analysis will be used within CEEDs at the following instances:

- Presenting the raw data [**CF1**]: The data has to be pre-processed if the property of the data (number of samples, dimension of the data …) does not allow representing the dataset as a whole, or if the specific research interest is related to specific **features** of the data that should be presented.
- Producing novel presentations based on input from the CEEDs Sentient Agent [**CF4**]

3.1 Steps of data analysis

Within CEEDs in principle any kind of data can be analysed. The methods used for the data analysis have to be adapted to the type of data. In general we can already distinguish between three phases of the analysis:

1. Pre-processing: Cleaning and artefact removal, dealing with missing data, outliers, etc.
2. Explorative data analysis: Necessary if no previous knowledge about any regularity in the data is available. In this step appropriate **features** of and/or **patterns** in the data have to be identified and analyzed.
3. Modelling the regularities in the data: Generating and validating **hypotheses** about the regularities in the original data, the features, or for specific patterns.

All three steps might be run through more than once because, for instance, what is considered as an artefact or an outlier might depend on the concrete understanding of the data.

In order to close the loop we have to generate meaningful hypothesis for the data, i.e. we need expert knowledge from the corresponding domain. Therefore we will focus in the following on the applications that will be developed within the project and for which specific domain knowledge is available within the consortium.

3.2 Application domains: Types of data and specific analysis methods

3.2.1 Archaeology

In the archaeology application the data are sherds of pottery collected from cites of ancient cities in Greece. One sample consists of a single sherd and the location where it was found. The data are:

1. Photographs or 3D scans of the single shards. The analysis of the first type of data will include
    a. Classification of the sherds according to the type and historical epoch.
    b. 3-D reconstruction of the pottery
2. Databases containing the location and several properties of the sherd. In particular the database contains the type of pottery, its function and the historical epoch of its origin. This information is already the result of a classification performed by the archaeologist. At the moment databases from two archeological sites, Koroneia and Tanagra are available from UL. The aim of the

data analysis is to build a statistical model of the pottery data base where we will use "topic models" [1] and "Markov random fields".

3.2.1.1 Sherd Classification (CERTH/ITI)

The Archaeology application in CEEDs involves classification and partial matching of archaeological sherds. In the first scenario, sherds are classified into groups of the same or similar relics. CERTH/ITI is currently working on a novel technique for automatic sherd classification. A description of the algorithm along with its first results is presented here.

Firstly, experts (archaeologists) define an indicative sherd from each class. The classification system is trained based on these indicative sherds. Front and back side images of each sherd are manually captured. For each sherd's side, a region of interest (ROI) is detected. Global and local descriptor vectors are then produced from sherds' regions of interest.

The Figure 7 illustrates a characteristic example of ROI extraction. In the original sherd (a), there is a number sign. Conservators often write numbers on sherds to help them during the classification process. Accurate ROI extraction can overcome such problems. Figure (a) is the original sherd, as it was captured by the camera, while figure (b) is its region of interest.

(a) (b)

Fig. 7 - *Region of interest extraction in an archaeological sherd: a) Original captured sherd, and b) region of interest.*

ROI extraction is used so as to improve the accuracy of the classification results. It is based on smoothing regions in each sherd, trying to estimate sherd's prevalent color and discard pixels that belong to degraded areas.

The ROI procedure can be summarized in the following steps:

Smoothing

To both sides of each sherd image a smoothing procedure is performed. That is, pixel values are changing according to the following equation:

$$p_{i,j} = \frac{\sum_{y=i-k}^{i+k} \sum_{x=j-k}^{j+k} p_{y,x}}{(2 \times k + 1)^2}$$

Where i,j are the pixel coordinates and k is a constant, which represents the neighborhood size defined for achieving image smoothing.

Local minima and maxima detection

Local minima and maxima are detected for both sides of each sherd image, within a constant local neighborhood area.

ROI extraction

According to the assumption that the number of local minima and maxima is increasing in degraded areas, around each local minimum and maximum there is non-useful information, which should be ignored. Each area is a circle with center the local peak (minimum or maximum) and range constant (k). The rest of the image pixels are considered as useful regions which are further processed to extract global and local features.

The following figure illustrates an example of a sherd's part with detected local minima (the pixel with the lowest value in a neighbourhood) and maxima (the pixel with the greatest value in a neighbourhood). All pixels inside non useful regions are rejected, while all other pixels consist of the region of interest.

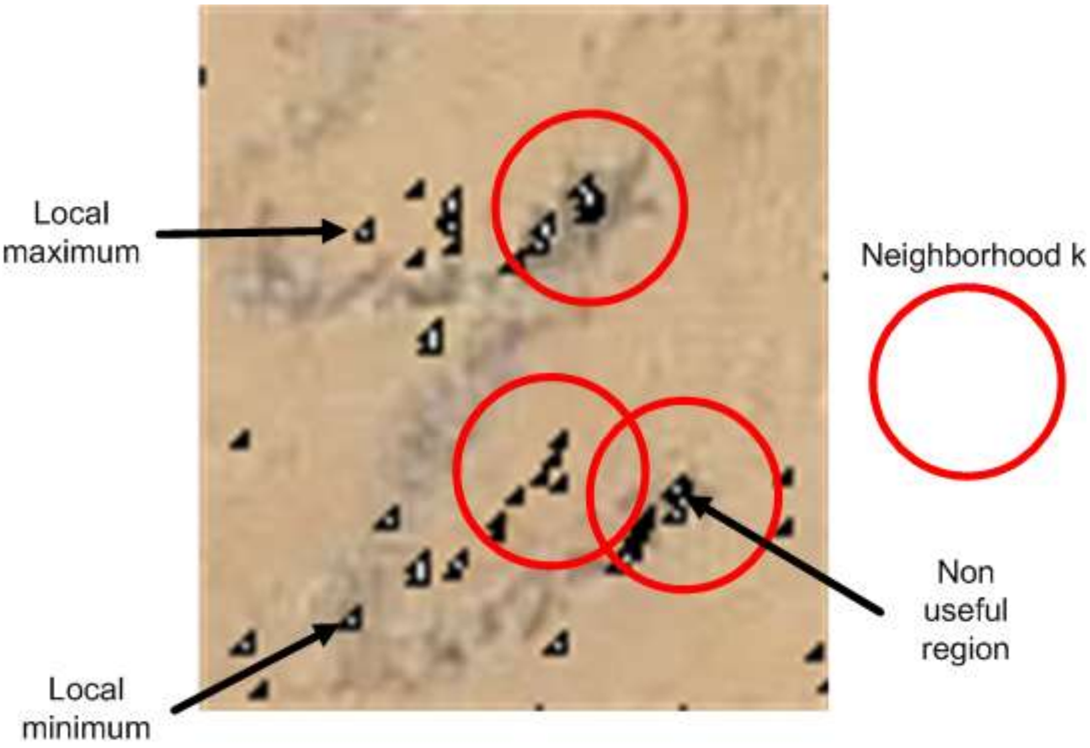


Fig. 8 - *Region of interest extraction based on local minima and maxima*

Therefore, pixels that belong to these areas should be ignored during the feature extraction process.

Feature vectors

For each side of each sherd image, two feature descriptor vectors are extracted.

In- Phase feature (global feature)

Since, the most representative feature for sherd classification according to conservators is chrominance, the color model selection is a critical decision.

YIQ and specifically the "I" component is chosen as the most appropriate color model to work with. The following equation is used to transform an image from RGB to In-phase:

$$Iph = [0.595716 - 0.274453 - 0.321263] \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

As it is shown, the red component is favored comparably to green and blue components. Taking into account the characteristics of in-phase, the following transformation, a generalization of the equation above, is created:

$$IphR = [0.595716 - 0.274453 - 0.321263] \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$IphG = [0.595716 - 0.274453 - 0.321263] \begin{bmatrix} G \\ R \\ B \end{bmatrix}$$

$$IphB = [0.595716 - 0.274453 - 0.321263] \begin{bmatrix} B \\ G \\ R \end{bmatrix}$$

IphR, IphG and IphB result from the application of the above transformation in each pixel of the image. That is, each pixel gives three output values, one for each component of the RGB color model.

For each resulted image ($IphR$, $IphG$ and $IphB$) the corresponding histogram is constructed. In most cases, histogram information is gathered in a relative small area inside each histogram. Therefore, histogram stretching, bin smoothing and normalization are performed.

Based on the procedure described above, IphR, IphG and IphB histograms are merged and a 1X765 global descriptor for each image is constructed.

Contrast feature (Local feature)

Local contrast can reveal useful information about the texture of the sherd, depending on its morphology. Additionally, contrast variation over the image can reveal warping or rills in the sherd.

Local contrast is computed for each pixel upon in-phase images (IphR, IphG and IphB) according to Michelson contrast, which is defined as:

$$p_{i,j} = \frac{\max[N_{i,j,k}] - \min[N_{i,j,k}]}{\max[N_{i,j,k}] + \min[N_{i,j,k}]}$$

The corresponding contrast histogram is constructed for each sherd and the following procedure is similar to the in-phase feature performing histogram stretching, bin smoothing and normalization.

KNN Classifier

Classification is achieved using the K-Nearest Neighbor classifier (KNN). The main reason for the selection of the KNN classifier is due to its simplicity and low computational complexity.

Experimental results

In order to test the classification outcome the following measures have been used:

- The **True Positive (TP)** rate is the proportion of sherds which were classified in a certain class, among all sherds which truly should be classified to this class. True positive is the same as recall.

- The **False Positive (FP)** rate is the proportion of sherds which were classified in a certain class, but belong to a different class, among all sherds which are not of this class.

- **The Precision** is the proportion of the sherds which truly belong to a class among all those which were classified in that class

- The **F-Measure** is simply 2*Precision*Recall/(Precision+Recall), a combined measure for precision and recall.

Database

In cooperation with the 9th Ephorate of Byzantine Antiquities in Thessaloniki, Greece, we created a database of 55 pottery sherds (front and back sides) that are classified in 6 classes. In the following table the number of sherds for each class are shown. For each sherd two photographs (110 in total) were taken, one for the front side and another for the back side. Photographs are taken with a camera from a fixed distance.

Tab. 1 - *Categories of pottery sherd images*

Category 1	23 sherds (46 captures)	Category 4	4 sherds (8 captures)
Category 2	11 sherds (22 captures)	Category 5	4 sherds (8 captures)
Category 3	8 sherds (16 captures)	Category 6	5 sherds (10 captures)

Results

According to the classification outcome, the proposed method performed:

Tab. 2 - *Classification results*

Ground truth sherds used for training	6	
Correctly Classified sherds	38	77.551 %
Incorrectly Classified sherds	11	22.449 %

The following table demonstrates the detailed accuracy for each class:

Tab. 3 - *Detailed accuracy by class*

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
0	0.864	0.111	0.864	0.864	0.864	0.876
1	1	0.026	0.909	1	0.952	0.987
2	0.857	0.024	0.857	0.857	0.857	0.917
3	0.333	0.087	0.2	0.333	0.25	0.623
4	0.333	0.022	0.5	0.333	0.4	0.656
5	0.25	0.022	0.5	0.25	0.333	0.614
Weighted Avg.	0.776	0.067	0.779	0.776	0.772	0.854

3.2.1.2 Shape based analysis of ceramics/pottery sherd profiles towards automatic information retrieval and guided exploration of taxonomic spaces

The approach we follow is complementary to others (in CEEDs) and is innovative. In a nutshell, the idea is to start from a single snapshot or a series of snapshots or a video shot taken by the archaeologist in situ of a sherd of interest (or set of sherds) such that a sherd profile is either visible or reasonably visible (i.e., that enough of a profile can be segmented to allow for a useful match even if partial). Then an image processing step attempts to segment the profile region automatically. A shape representation is computed to derive shape features useful in retrieving the classification match from archaeological taxonomic references based on profiles of ceramics, pots, sherds.

There is a long tradition in archaeology of using single partial profiles of ceramics and pottery to classify these objects; the main advantage being that if the sherd captures enough shape information a robust match can be made with high confidence.

Large volumes of profiles have been compiled over the years by field archaeologists and these books represent a goldmine of shape-based information we want to exploit in the context of CEEDs. Since many field archaeologists are already familiar with such references, the interactions between CEEDs and the expert user should prove engaging and provide for a different mode of narrative to other means of information retrieval, where the narrative space can be rooted in the reference book itself: i.e. permitting an informed navigation of these large books with guidance, help, provided by CEEDs. Rather than having the archaeologist flip through the pages and regularly refer to an index, a cumbersome task, CEEDs can visually and interactively guide the expert in either validating the matches provided by CEEDs, or refine these using the most relevant book entries as a navigation guide.

A typical such taxonomic reference (relevant to the site the Leiden team is working at) is:

1. Hellenistic Pottery: Athenian and Imported Wheelmade Table Ware and Related Material. Part 2: Illustrations. Author(s): Susan I. Rotroff. Source: The Athenian Agora, Vol. 29 (1997).

We plan to fully characterise the shape features of profiles found in such a book which are already been scanned in electronic formats to buil templates and indexed references of ground truth (Task A).

Each time a new sherd is captured from a photo/camera shot by the user, or uploaded from a database of collected snapshots, we provide a similar algorithmic method to produce a shape index which can be used in finding a match from the reference book(s) (Task B).

Task A can be conducted a priori (off line). It requires developing segmentation techniques adapted to the analysis of black and white book pages as is typical from the archaeology taxonomic references. Ideally, this integrates some existing OCR technique to automatically associate the right meta-data (explicitly available on a book page, in the text of caption for the profile templates).

Task B is conducted on the fly whenever a new input image is provided. This typically require processing colour images and evaluating the best possible profile extraction, when dealing with a video sequence or set of shots for a given sherd found in situ and of interest to the user.

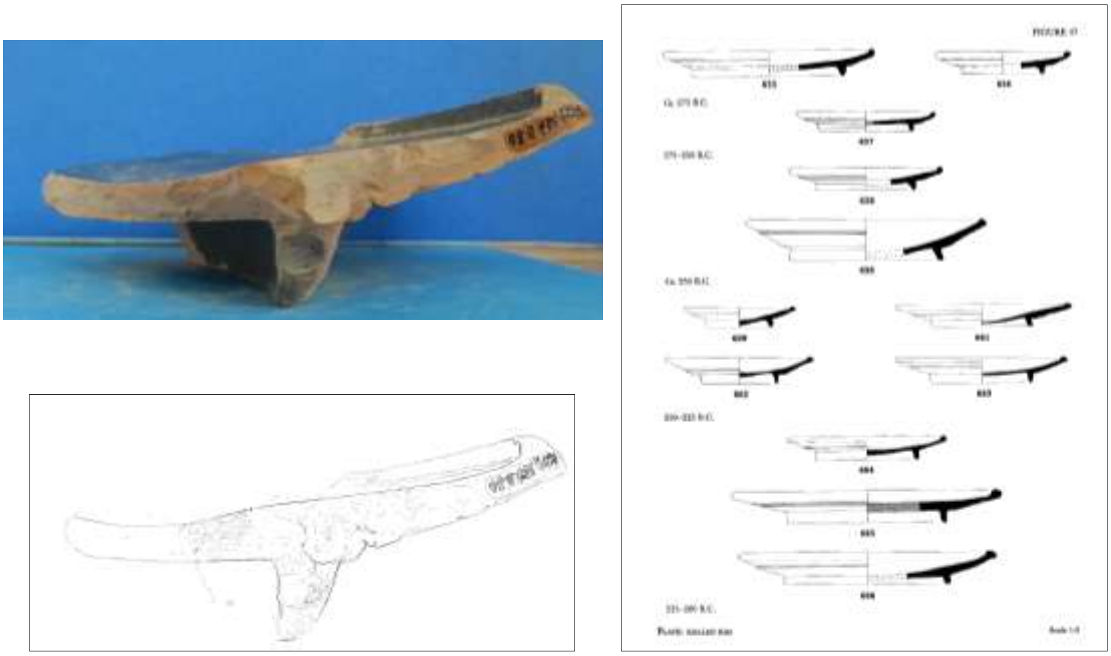


Fig. 9 - *Profile's segmentation and matching with likely profiles match (templates) found in reference book.*

A. First implementation reporting

In a first attempt to produce a useful shape indexing mechanism for profiles of ceramics and pottery, we have planned to exploit shock graphs: 2D graph-based representations of distance from outline symmetries, an extension of the classical medial axes and Voronoi diagram widely used in the computer vision and computational geometry fields for characterising the shape of 2D objects. This is a topic in which we have developed an expertise over the years (see short bibliography below) and thus plan to extend its application to this particular problem.

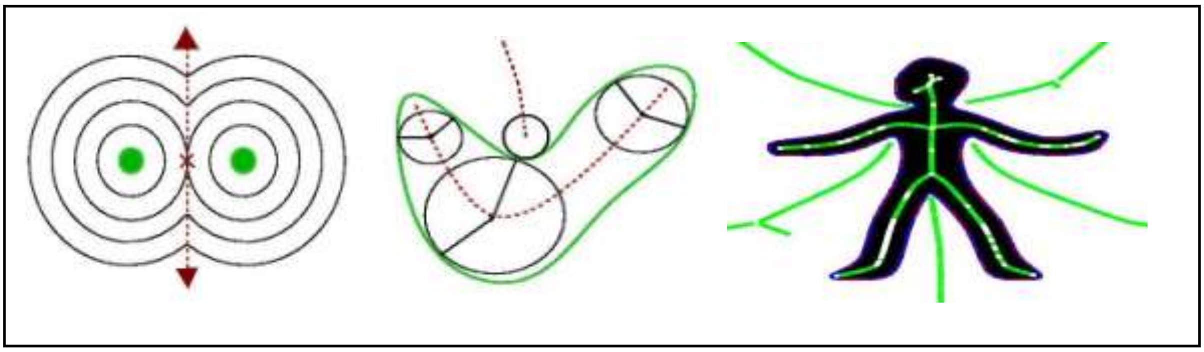


Fig. 10 - *Concept of a shock graph. The Medial Axis of Blum: (a) for two sample points as the locus of meeting Euclidean wavefronts, where the cross indicates the initial shock when the fronts first meet, and the arrows indicate the direction of growth or flow of the medial segments; (b) as the (equivalent notion of) loci of centres of maximal contact disks (i.e. such centers are where shocks are formed); (c) a computed shock graph for a humanoid 2D binary object; both an interior an exterior symmetry graph is defined by the shock based formations: either or both can be used in information retrieval tasks.*

Fig. 11 - *Example of snapshots performed in situ by the Leiden archaeological team in consultation with Goldsmiths' team. Notice a piece of plaster is occasionally used to stabilise the fragment and the orientation is near optimal although performed without the need for professional photography expertise and equipment (also, some shadows are left).*

Together with the Leiden team we have started a trial scenario where the archaeologist captures in situ snapshot of sherds which make the sherd profile apparent: i.e. such that the sherd is oriented in the image plane similarly to the typical oriented templates found in taxonomic references. Essentially the sherd should be positioned an up-right position as if it were part of a complete and intact ceramic or pottery artefact. This is only achieved approximately by having the archaeologist rapidly position the selected sherd of interest (yet to be identified) on top of a piece of gum or plaster (preferably of a different color from the ceramics' material). To further help in the following stage of image segmentation, the sherd is position with a “blue screen” background. This can easily be achieved by using a piece of white or single color cloth. Ideally, bright and diffuse lighting is used to minimise shadows. The practical goal here is to ensure the set-up can easily be done in the field by an archaeologist with no special equipment required other than a basic digital camera.

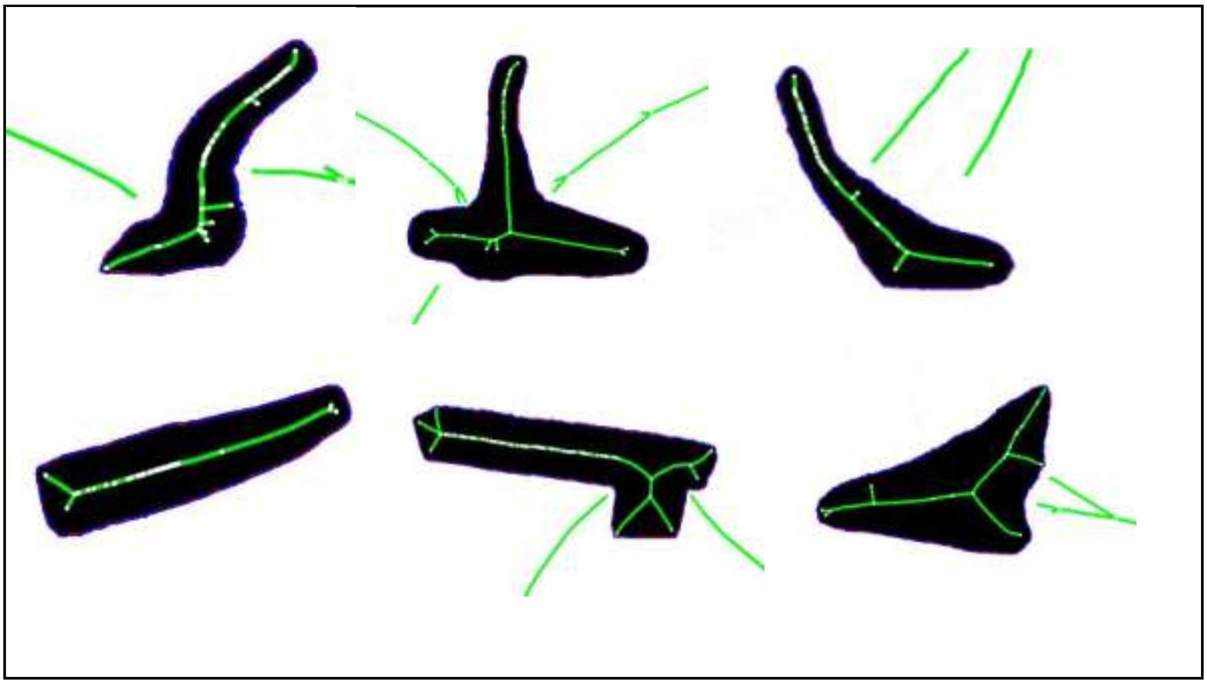


Fig. 12 - *Computed shock graphs for the inputs of Fig. 3 above. From these graphs we extract a shape-based index useful for information retrieval, e.g. by using corner locations and significance (based on curvature and relative area weight), main geometric global description (codon-like description as a sequence of convexities and concavities), branching structure (related to parts description).*

We have specifically developed first versions of automatic segmentation via binarisation (thresholding) of the in situ images, have adapted our shock graph computations to emphasise the most significant features (shock graph branches) useful for shape indexing.

Currently we are developing the complementary testbed to process the ground truth information available from the archaeology taxonomic reference book of relevance to the site for the Leiden team. Once this is in place, we will next need to develop an adapted shape matching method to find the best relations between shape encoded templates from the reference book with the indices derived from images collected in situ. This should then provide a sufficient basis to test our CEEDs scenario and explore new narratives related to taxonomy explorations.

To the best of our knowledge, this is the first time a shape indexing method is derived from shock graphs in this manner. Also, to the best of our knowledge, this is the first time the classical archaeological taxonomic resources of profiles for ceramics and pottery is being exploited explicitly in an automatic information retrieval process.

3.2.1.3 Statistical Modelling of the Pottery databases (MPG)

Each entry in the pottery databases corresponds to one classified sherd and contains

1. the location, i.e. the cell in the grid, where it was found
2. the type of pottery, such as cup, plate, amphora, …
3. the time period, such Hellenistic, early roman, …
4. the function of the shard, such as storage, transport, architecture, …
5. further entries that will be not considered in the analysis

The type of the pottery is the outcome of the classification procedures described above. The function of the sherd (that should not be confused with the function of the area that will be used below) is mainly inferred from the type by the archeologists and will be not considered in a first analysis.

For the statistical analysis it is assumed that each entry is a sample randomly drawn from a joint probability distribution. As a starting point we restrict ourselves to the joint probability distribution p(y,x,t) of the random variables Location (X), Type (Y) and Time (T). This distribution quantifies how likely it is that a randomly chosen sherd of type y was found in location x and is assigned to the time t.

The statistical model will explain this joint probability distribution by assuming an additional hidden variable f that corresponds to the functions of areas of the ancient city that are up to now inferred by archeologists from the spatial distributions of the different types more intuitively. The basic idea is that a certain function of an area at a certain time generates a specific distribution of types of pottery, p(y|f) in that area. In the general case this distribution is time dependent, i.e. p(y|f,t). The joint probability distribution p(y,x,t) is then modelled as $p(x,y,t)=\Sigma_f\, p(y|f,t)p(f|t,x)$. For fixed time t this leads to a statistical model that is known as "Topic model" in the context of text analysis. There X, our location, corresponds to the texts and Y (type of pottery) to the words, while F, the area function in our case, represents the different topics. Using this kind of models is an innovative approach in ths domain. As a first step MPG implemented a maximum likelihood estimation of such models using an expectation maximization (EM) algorithm. Fig.?? shows a result for the Tanagra data base. One can see that most types of pottery are assigned to only one function by the model and thus could be used to identify the function – provided the model would be correct. However, these are only preliminary results because the EM algorithm might find only local maxima of the likelihood and maximum likelihood estimates in general will not allow to assess the robustness of the assignment of a function to a location. Therefore no interpretation is given for the different functions at this state. MPG is working on a Bayesian estimation of the model where visualizing the sampling of the posteriors for p(y|f) and p(f|x) (which are distributions over distributions) will allow to judge the reliability of the assignment.

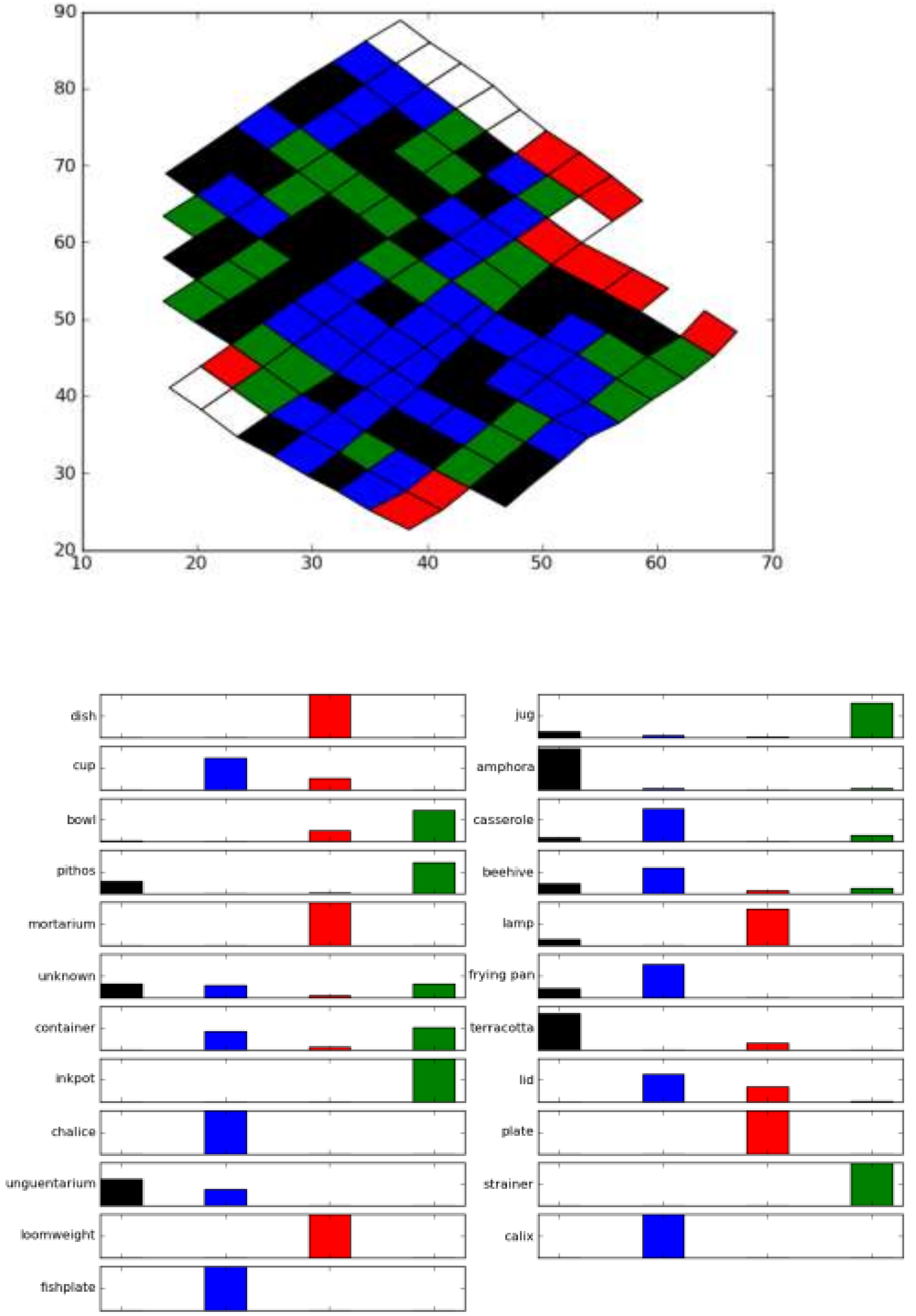


Fig. 13 - *Maximum likelihood estimate of a topic model for data from the Tanagra data base, hellenistic – early roman time, using the EM algorithm with 4 functions. Top: Shows the color code of the function for which p(f|x) is maximal for each location x. Bottom: p(f|t) for the different types t.*

3.2.2 Neuroscience

In this application domain we can distinguish first between **experimental** data and data from **simulations**.

Experimental data can be only **structural data** – either anatomical or functional or they can also include **temporal** information.

Examples for structural data are the **connectome data**: Matrices representing the structural and functional connectivity of neural networks. In particular we are looking at the connectom of the human brain (data from O. Sporns) with 66 regions being the nodes and numbers between 0 and 1 representing the coupling strength between the regions. Structural data can be visualized as weighted graphs and explored using graph theoretical measures such as degree distribution, cluster coefficient, or assortativity. For the connectome data we plan to implement simulations in iqr using the information of the structural connectivity matrix and producing a functional connectivity matrix. This can be used to test hypothesis about the direction of the coupling which cannot be measured directly and is therefore not contained in the structural connectome data.

Examples for data with a **temporal structure** are neural time series such as Electroencephalogram (EEG) or Magnetoencephalogram (MEG) data, local field potentials (LFP) or spike train data from multi electrode arrays (MEA). Available data sets include EEG data from coma patients (UPF) and sleep EEG data (Universität Zürich/MPG).

For the exploratory analysis either the raw data or certain features of the data will be either visualized and/or certain features will be transformed into sound patterns (**sonification**). Features that will be used include the spectral power in different frequency bands, amplitudes, phase and mean frequency in certain frequency bands. Parameters from time frequency analyses (linear models with time dependent parameters e.g. adaptive autoregressive models, short term Fourier transform, wavelet analysis, matching pursuit algorithm) will be used both for the definition of features and pattern detection.

For the analysis of data from **simulations** we want to implement real time calculation of **information theoretic quantities** such as mutual information, multi-information, conditional mutual information to enhance the understanding of simulations in neural models of cognitive processes. Starting point will be the cerebellum model at UPF that is implemented in iqr and already visualized in CXIM.

3.2.3 Appliance

The appliance application scenario will explore 3D cad model data from Electrolux appliances. Except from the raw 3D data the similarity search engine will produce descriptors metadata for each 3D model for use in the search procedure.

In the case of global features the engine will produce one descriptor vector for each model. The length (size of the vector) and the data type of the vector depend on the descriptor extraction algorithm.

in the case of local feature descriptors, each model may be represented by a large number of descriptor vectors one for each saliency area on the 3D model.

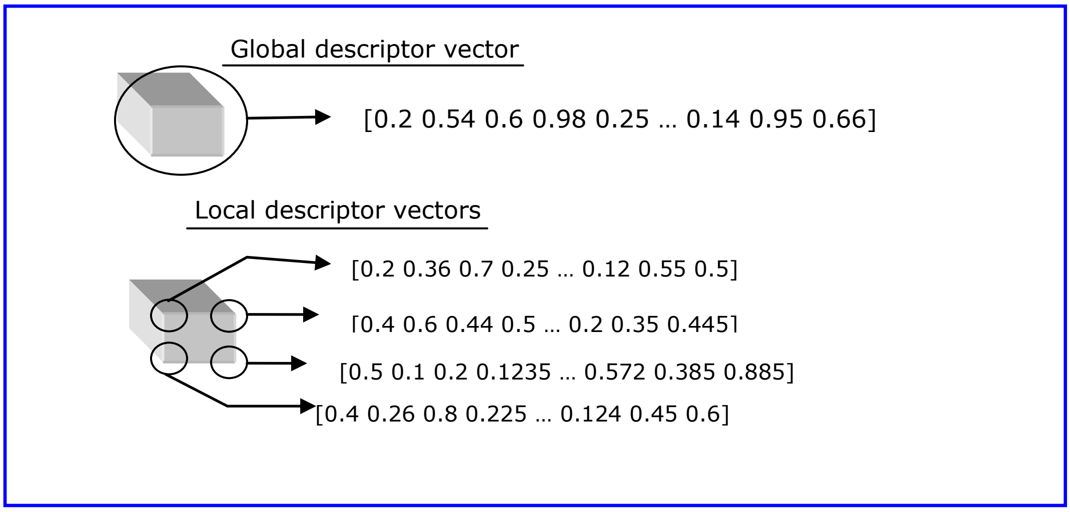


Fig. 14 - *Diagram showing global vs. local feature extraction approach*

CERTH/ITI currently works on developing a new method for 3D local descriptors which is based on saliency analysis of the 3D models.

4 Narrative Generator

The narrative generator defines how the information is presented to the user. This is the main channel through which the CEEDs Sentient Agent is interacting with the users.

The implementation of the narrative generator will be tackled in parallel by two different, independent approaches, with a subsequent integration. The first approach is using real-time generation of event sequences, based on heuristic search planning and/or local search implemented by TEESIDE and described in the next sections, while the second approach aims at the application of the Distributed Adaptive Control (DAC) architecture for sequence learning implemented by UPF.

4.1 Based on DAC (Distributed Adaptive Control)

The approach based on DAC means to devise a meaningful plan for constructing the narrative engine component based on empirical evidence with specific goals. Thus, within the specific application scenarios, experiments are planned in order to define a systematic way of organizing the narrative according to the CSA outputs. This calls for an iterative process, where the more the input and output loops in the CEEDs engine are developed, the more sophisticated the narrative structure rules can become. This approach is firstly implemented in the Bergen-Belsen Memorial application prototype, as described in more details in Deliverable 6.1. To sum up, in this first iteration, the historical data is organized in stories (thematic units based on the Bergen-Belsen museum exhibition organization) that are randomly assigned at runtime to a specific placeholder in the virtual world (a different barrack every time). The first step we subsequently took was, using this framework of stories, to find an optimal order for the presentation of data in terms of recollection. Once our experiments defined this optimal path for guiding the users, we compared it with an active exploration of the space, and empirically tested our hypothesis that a form of guidance enhances recollection as opposed to a free navigation in the virtual space. The results from this study will help us built a more sophisticated version of the narrative component of our application.

4.2 Based on Heuristic Search Planning

4.2.1 Overview

The narrative generator composed of a Heuristic Search Planner (HSP) uses propositional representation of the world and user (i.e. world and user states) in order to generate a sequence (i.e. a plan) of state transition functions (i.e. operators) in order to reach a goal state.

The narrative generator is implemented in C# with the .Net 4.0 framework and is constituted of 4 different projects:

- *HSP Syntax*: a dynamic library (DLL) that holds all the HSP planning syntax and planning algorithm used by the narrative generator.
- *Narrative generator - wrapper*: a DLL providing all the high level functionality of the narrative generator and can directly be included within application.
- *Narrative generator - run-time*: application that can be integrated within CEEDs engine by communicating using YARP network technology.

- *Authoring Tool*: an application which allows editing and manipulating PDDL files that are needed by HSP to produce plans.

4.2.2 Heuristic Search Planning and Planning Syntax

Heuristic Search Planning (HSP) is a state-of-the-art planning (Bonet & Geffner, 1999) that has immediately demonstrated great potential, even outperforming many other planning techniques on traditional planning benchmark problems.

Our HSP implementation uses a Stanford Research Institute Problem Solver (STRIPS) based representation for problem description (Fikes and Nilsson, 1971), and searches the space of states from the initial state, using a traditional heuristic search algorithm and heuristics automatically extracted from the STRIPS formulation. In simple terms, these heuristics measure the prospect of reaching the goal state from each of the new generated states, by computing the length of an operator chain that would generate the propositions composing the goal state.

In order to produce a sequence of actions that leads from the initial state to a state matching the goal, a HSP planner takes three inputs, all encoded in a formal language like STRIPS: i) a description of the initial state of the world, ii) a description of the desired goal, and iii) a set of possible actions.

The HSP process uses a real-time search algorithm (RTA*) to assess and select the best next available operator. Operator selection is performed using a heuristic value computed for the corresponding state.

The HSP syntax implementation is composed of 11 classes that can be categorised into 3 main categories:

- Knowledge representation: constituent of the planning domain and problem representing a STRIPS instance.
  - *Predicates*: Adapted from a set-theoretic representation, which relies on a finite set of proposition symbols, predicates are intended to represent various propositions about the world.
  - *Operators*: Planning operators are state transition functions composed of i) preconditions (i.e. propositions that must be established before the action is performed) and ii) post-conditions/effects: propositions that is established after the action is performed
  - *States*: Planning State holds the representation of the world and is composed by a set of predicates that are stored using a sorted dictionary that is used as a Hash table in order to improve efficiency. Several heuristic computation methods are implemented (VI, IVI, BF and PINCH). For more information about the computation algorithms, consult the reference (Liu, Koenig, & Furcy, 2002).
- Planning Syntax: represent inputs of planner (i.e. planning domains and planning problem).
  - In automated planning, the planning domain holds the list of predicates and operators that are used to solve any given planning problem.
  - The planning problem, on the other hand, is the tuple initial and goal state that are used to generate the sequence of operators.
  - The planning domain and problem are loaded from standard Planning Domain Definition Language (PDDL) files.
- Search Algorithm: Different syntax and algorithms to find solutions within generated search tree. Two different methods of search are implemented, an offline and an online method:

- Offline method represent simple A Star search
- Online method provides real time version of A Star (RTA Star) in order to cope with replanning and dynamic environment.
- The two different search algorithms follow the ones described in (Korf, 1990).

4.2.3 Narrative Generator

The role of the Narrative Generator (NG) is to generate sequence of actions from the representation of user and world state, which are updated by the CEEDs Sentient Agent (CSA), in order to be dramatized by the Composition Engine (CE).

The two different versions of the NG (wrapper and runtime) offers high level functionalities such as the loading of planning syntax, the generation of sequences of actions to be performed by the composition engine, the updating of the world representation (i.e. used to integrate input of the CSA) or even the function used to signal that actions have finished being executed).

In the wrapper version of the narrative generator those function can directly be called by any .NET application. However, the run-time version of the narrative generator acts as a sandbox and uses YARP network messages to communicate with the CSA (as input) and the CE (as output).

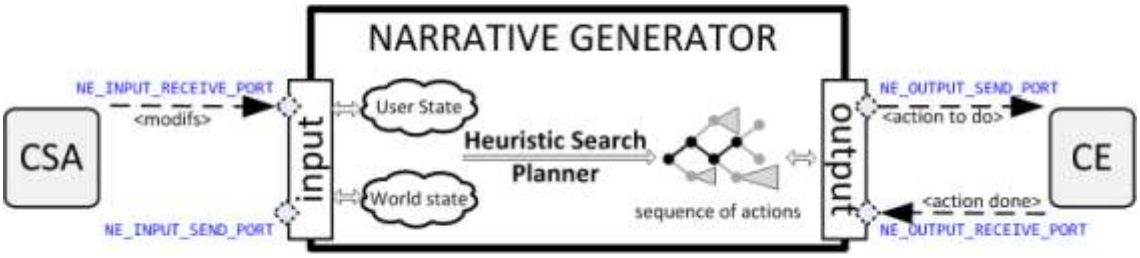


Fig. 15 - *Integration of the narrative generator run-time within the YARP Network*

Each communication interface of the NE (input and output) uses two YARP ports: one for sending message and one for receiving messages (see Figure 15).

The input interface is in charge of communicating with the SA in order to maintain and up-to-date copy of the word and user state. The messages are received from the CSA via the `NE_INPUT_RECEIVE_PORT`.

Message can be sent in the form of: (TO DETERMINE/CONFIRM)

```
:predicate PRED :value VALUE
```

With `PRED` = name of the predicate to change and `VALUE` = its new value. In that format, if a predicate is already present in the state its value will be updated otherwise. We have to determine if feedback have to be sent to the CSA.

The output interface of the NG is in charge of communicating with the CE in order to dramatize the actions and influence the user experience. Once planned according to the user and world states, using Heuristic Search Planner (HSP) the NG produces a sequence of actions that are sent to the CE in order to be executed.

The *<action to do>* are sent to the CE via the `NE_OUTPUT_SEND_PORT` with the following format:

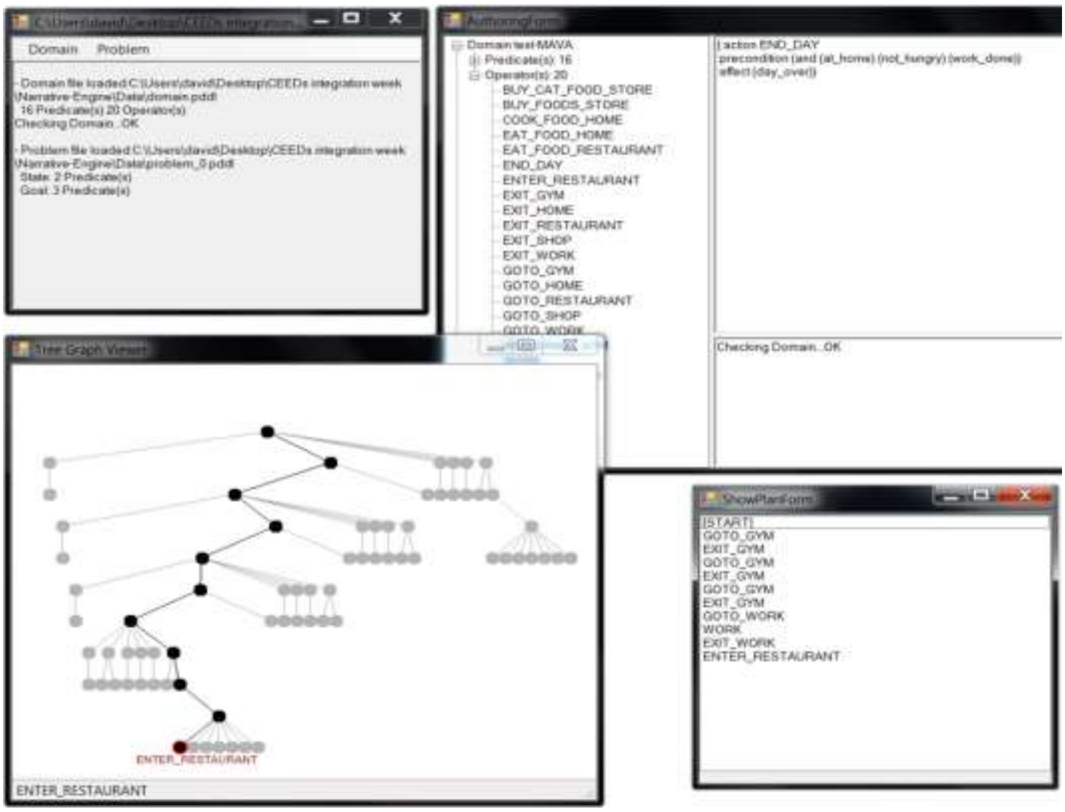
```
:action NAME :state STATE
```

With `NAME` = name of the action to execute and `STATE` = state of the action which can take value ∊ {`TODO, DONE`}

Once the action has been executed by the CE, it is sent back to the `NE_OUTPUT_RECEIVE_PORT` in order to move to the next action.

4.2.4 Authoring Tool

The authoring tool (Figure 16) allows both the creation/edition of planning domains and problem in the form of PDDL files, and also their validation/testing through the use of offline planning and manual exploration of the resulting state-space tree generated.



```
- Domain file loaded C:\Users\david\Desktop\CEEDs integration week
\Narrative-Engine\Data\domain.pddl
 16 Predicate(s) 20 Operator(s)
Checking Domain ..OK

- Problem file loaded C:\Users\david\Desktop\CEEDs integration week
\Narrative-Engine\Data\problem_0.pddl
 State: 2 Predicate(s)
 Goal: 3 Predicate(s)
```

```
(:action END_DAY
precondition (and (at_home) (not_hungry) (work_done))
effect (day_over))
```

```
Checking Domain ..OK
```

```
[START]
GOTO_GYM
EXIT_GYM
GOTO_GYM
EXIT_GYM
GOTO_GYM
EXIT_GYM
GOTO_WORK
WORK
EXIT_WORK
ENTER_RESTAURANT
```

Fig. 16 - *Snapshot of the authoring tool*

The elementary PDDL file I/O manipulations include functionalities such as Load/Save Domain files in PDDL format, and Add/Delete/Edit Planning elements (propositions, operators, state, and goal). In addition, we included syntactic integrity checking which procedurally control the consistency planning elements and some basic content integrity for heuristics computation.

As we are likely to deal with large dataset, the authoring tool also proposes visual search tree exploration with i) manual expansion of nodes and ii) heuristics computation in order to help the knowledge engineering process.
Finally, we integrate some Planning engine to simulate run-time using different searches (Offline and Real-time version of A Star search).

5 CEEDs Sentient Agent (CSA)

The CEEDs engine orchestrates the interaction between the users and the data space that will be explored. The internal processing of the engine is based on three main components: Firstly, adaptive data mining and discovery that defines what data is presented to the user, secondly, spatio-temporal structuring of the presented content in the form of narratives generated by the composition engines, and thirdly, the "CEEDs Sentient Agent" (CSA).

The CEEDs Sentient Agent will be equipped on one hand with the internal model of the user, and on the other hand will have its own interests, intentions, and personality. This agent will act as an independent component in the architecture, receiving the inputs (implicit and explicit signals of user) and sending messages to other components like the narrative generator to influence which content is delivered to the user.

The CEEDs Sentient agent adapts the user model for each CEEDs application scenario and will be integrated into the engine in a second stage with the implementation of CF4/CF5.

The CSA will be developed based on the implementation of "Ada: The intelligent space" (K. eng, et all 2003) and will profit from the entire audiovisual equipment of the XIM (Bernardet et al. 2008), while its behaviour will be constantly tuned to the real-world variables such as the user's position and response, to achieve a multi-modal rendering of perceptive cues and control the environment parameters.

5.1 Development of the CSA

5.1.1 First stage prototype

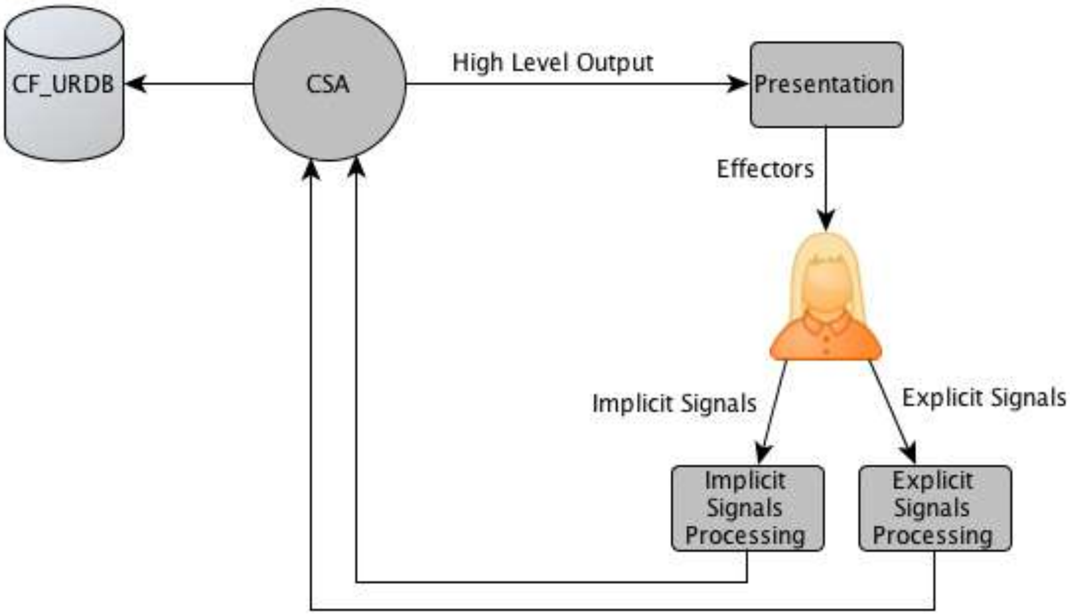


Fig. 17 - *Scheme of the first stage of development of the CSA*

The CSA will be implemented in the XIM and will feature a decision making system which optimizes the data presentation according to the user's preferences and reactions. It will interact with the user who explores the space and generate a set of predictions to present the most effective subsets of the data, types of data presentations and interactions that modulate different states of the creative process.

On a first stage a prototype of the CSA will be developed upon a closed loop interaction between the user and the CSA (Figure 1). The CSA will present high level output to the user, this output will be continuously adapted according to the implicit and explicit signals received from the user. The user responses are recorded and stored in the User's response database (CF_URDB).

5.1.2 Second stage prototype

The second stage of the development will consist of the integration of the CSA in the CEEDs core engine. During this second stage, the architecture of the CSA will be further implemented and organized in four layers following the DAC architecture (Verschure et al. 2003):

- Reactive layer: Direct mapping of user behaviour to the control of effectors;
- Adaptive layer: Output adapted to the past and present behaviour of one or more users;
- Contextual layer: Ability to create novel interactions based on the integration of contextual information (both in time, across users, and across data domains);
- Reflective layer: The top layer (reflective) implements the highest level of autonomy of the CEEDs engine, where the CEEDs Sentient Agent can reflect on its own internal workings and behaviour and use this information to develop novel methods for discovery.

6 Work package activities

6.1 CEEDs Integration Meeting

During the reporting period UPF organized and hosted a technical Integration Meeting in Barcelona, Spain from June 06 to 08 2011. The meeting was attended by technology providers (WP2), application (WP6) and engine developers (WP3) and integration partners (WP5), and served to discuss the overall architecture of the CEEDs Engine, applications and to do hands-on work, making first attempts to integrate the technologies that will be used in the project.

During the first day the CEEDs core features, the CEEDs engine, and available technologies were presented. On the second and third day attendees split in groups for focus discussions around applications development and the integration of different technologies. In details here is presented an overview of subjects discussed during the meeting:

- Introduction applications and use cases
    - Overview CEEDS applications, core features, use cases (GOLD)
    - Overview general CEEDS application architecture (UPF)

- Architecture of CEEDS applications
    - Per application
        - Introduction application
        - Architecture of the application (inputs from WP2, outputs to WP4, integration requirements (WP5))
- Applications:
    - Appliance (UH)
    - Science (neuroscience) (UPF)
    - Archaeology (GOLD)
    - Historic (Bergen-Belsen) (UPF)
- Frameworks
    - Narrative framework (TEES)
    - Bergen-Belsen application framework (UPF)
    - XIM architecture (UPF)
    - Distributed communication and YARP (UPC)
    - Social Signal Interpretation (SSI) framework (UAU)

This event was important to present the design of the CEEDs engine and to receive inputs from partners to improve the design. It was also a very important to do practical advances on the integrations of tools and algorithms that are being develop in the CEEDs applications.

7 References

[1] Bonet, B., & Geffner, H. (1999). Planning as Heuristic Search: New Results. *European Conference on Planning, ECP'99*, (pp. 360-372).

[2] Eng, K., Klein, D., Baebler, A., Bernadet, U. , Blanchard, M., Costa, M., Delbruck, T., Douglas, R.J., Hepp, K. , Manzolli, J., Mintz, M. , Roth, F., Rutishauser, U., Wassermann, K., Whatley, A.M., Wittmann, A., Wyss, R. & Verschure, P.F.M.J. (2003) Design for a brain revisited: The neuromorphic design and functionality of the interactive space Ada. Reviews in the Neurosciences. 14: 145-180

[3] Fikes, R., & Nilsson, N. (1971). STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Second International Joint Conference on Artificial Intelligence*, (pp. 608-620).

[4] Griffiths, T.L. and Steyvers, M., Finding scientific topics, PNAS, 101(2004),5228-5235.

[5] Korf, R.E. (1990). Real-time heuristic search. *Artificial Intelligence, 42:2-3*, (pp. 189-211).

[6] Leymarie, F.F "Thoughts on Shape," in "Visual Thought: The Depictive Space of the Mind," L. Albertazzi, ed., Vol. 67 of "Advances in Consciousness Research" series, pp. 303-350, John Benjamins, Amsterdam, December 2006.

[7] Liu, Y., Koenig, S., & Furcy, D. (2002). Speeding Up the Calculation of Heuristics for Heuristic Search-Based Planning. *Eighteenth National Conference on Artificial Intelligence, AAAI2002*, (pp. 484-491).

[8] Rotroff, S.I. . "Hellenistic Pottery: Athenian and Imported Wheelmade Table Ware and Related Material." Part 2: Illustrations. Source: The Athenian Agora, Vol. 29 (1997).

[9] Siddiqi, K. & Pizer, S.M., "Medial Representations --- Mathematics, Algorithms and Applications," eds., Springer, volume 37 of Computational Imaging and Vision series, 2008.

[10] Verschure, P.F.M.J., Voegtlin, T., & Douglas, R.J. (2003). Environmentally mediated synergy between perception and behaviour in mobile robots. Nature, 425, 620-624.