Co-funded by the Horizon 2020
Framework Programme of the European Union

HORIZON2020 FRAMEWORK PROGRAMME

ICT – 21 -2014

Advanced digital gaming/gamification technologies

ProsocialLearn

Gamification of Prosocial Learning

for Increased Youth Inclusion and Academic Achievement

D3.3 2nd Prosocial affect fusion and player modelling

Document Control Page

WP/Task	WP3 / T3.2 and T3.3
Title	D3.3 2nd Prosocial affect fusion and player modelling
Due date	31/5/2016
Submission date	24/08/2016
Abstract	This deliverable is D3.3 2nd Prosocial Affect Fusion and Player Modelling of the ProsocialLearn (PSL) project 644204. This is the second version describing game technology capabilities for observing and analysing the performance of players aiming to learn prosocial skills using digital games in schools. The deliverable covers the Player Affect Observation and Learning Analytics subsystems of the PSL architecture.
Author(s)	Lee Middleton (ITINNOV), Simon Crowle (ITINNOV), Ken Meacham (ITINNOV), Michael Boniface (ITINNOV)
Contributor(s)	Kostas Apostolakis (CERTH), Kosmas Dimitropoulos (CERTH), Athanasios Psaltis (CERTH), Spyridon Thermos (CERTH), Kyriaki Kaza (CERTH), Kyriakos Stefanidis (CERTH), Stefano Modafferi (ITINNOV), Laura Vullier (UCAM), KAM Star (Playgen), Athanasios Psaltis (CERTH)
Reviewer(s)	Christopher Peters (KTH), Kosmas Dimitropoulos (CERTH)
Dissemination level	□ internal ☒ public □ confidential

Document Control Page

Version	Date	Modified by	Comments
0.0	4/5/2016	lee middleton	Initial version
0.1	19/5/2016	lee middleton, Michael Boniface	Move to skills view
0.2	26/5/2016	Athanasios Psaltis	Included fusion section
0.3	18/7/2016	lee middleton, Simon Crowle, Michael Boniface	Updates based on new skills approach
0.31	27/07/2016	Michael Boniface	Review, executive summary and glossary
4.0	03/08/2016	Michael Boniface, Kosmas Dimitropoulos	Addressing all reviewer comments and incorporating additional CERTH contribution on face and engagement observation
1.0	24/08/2016	Pilar Pérez, ATOS	Final and format review

List of Abbreviations and meaning of the most common use words

Abbreviation	Description
API	application programming interface
FOAF	Friend of a Friend (ontology)
HTTP	Hyper Text Transfer Protocol (s)
IRI	Internationalized Resource Identifier
JSON	JavaScript Object Notation
LRS	Learning Record Store
PLS	Prosocial Learning Specification’
PSL	ProsocialLearn
UUID	Universally unique identifier
xAPI	Experience API’

Executive summary

This deliverable is D3.3 2nd Prosocial Affect Fusion and Player Modelling of the ProsocialLearn (PSL) project project 644204. This is the second version describing game technology capabilities for observing and analysing the performance of players aiming to learn prosocial skills using digital games in schools. The deliverable covers the Player Affect Observation and Learning Analytics subsystems of the PSL architecture. This version extends the 1st version from Nov-15 (M10) until May-16 (M17) defining a Prosocial Learning Specification using xAPI and elaborating player affect fusion for emotion based on valance arousal, whilst outlining approaches for decision level fusion of multi-modal observations.

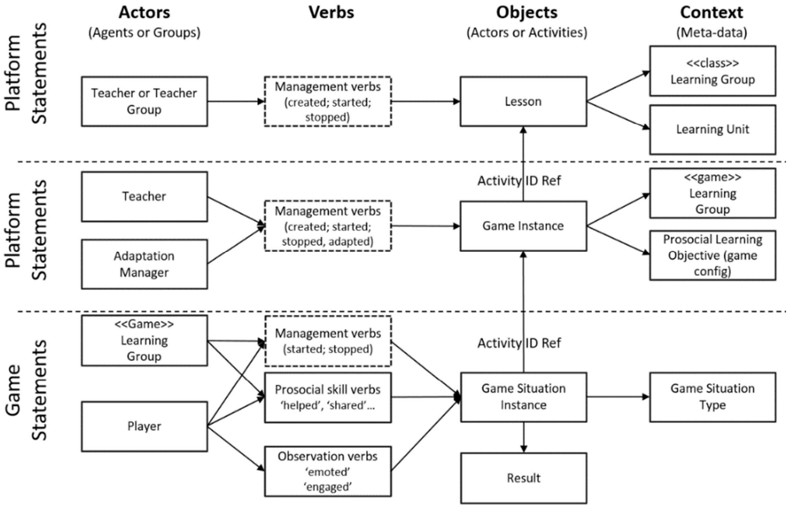


Figure 1: Player Model

We describe a player model used to store and analyse students prosocial behaviours and emotional and engagement responses during game play. We consider two orthogonal data sets in the representation of the user: the *student profile* and the *player history.* The former characterizes aspects of the student's identity and background information (in the context of the relationship to the learning provider) and psychological characteristics. In the latter case, the player history data records prosocial, emotional and engagement behaviors as well learning outcomes (and related data) captured during game play. The student profile data is anticipated to serve two main purposes. The player history elements of the model encapsulate observations made directly or indirectly (via a classification process) of players' behaviour during game play. Historical data will be used for three purposes: (i) as input to the PSL game adaptation algorithm, (ii) as information to be used by teachers in the assessment of students' progress and (iii) as data for experimental analysis. The player history is written using Prosocial Learning Specification (PLS) statements using extensions to the Experience API standard. The PLS information model is shown in the figure below. The specification allows the assertion of lesson and game context, emotion/engagement observations and prosocial skill scores.

We describe multiple modal player affect observation channels used to observe emotion and engagement. A series of multi-modal observation channels are established from input sensors connected to player devices including microphones, cameras and mouse/touchpad. Using sensing and classification techniques emotion from voice, facial expression and body language is be acquired and then fusion processes applied to provide temporal emotional state. We use a valance-arousal

space in order to measure emotion in all input modalities. This common representation allows us to compare and contrast the measures of emotion coming from different sensors. This is important if we want to perform data fusion. Individual sensor data is classified in the valence-arousal space via pre-processing and classifiers. After emotion classification, decision level fusion is employed. This serves to bring together the differing measures of emotion and find a best estimate for a given time instant. This fusion process happens in the valance-arousal space. The deliverable will be updated at PM25 (Dec-15) to describe the final version of the components.

Index

1 Introduction

This section provides detailed information about the purpose, scope and structure of the document as well as the intended audience of the document.

1.1 Purpose of the document

This document is D3.3 “2nd Prosocial affect fusion and player modelling” of the ProsocialLearn project 644204. The document describes game technology capabilities for observing and analysing the performance of players aiming to learn prosocial skills using digital games in schools. The capabilities include emotion and engagement affect observation and fusion, game interaction monitoring, social learning analytics, and visualisation feedback to teachers. The capabilities are combined to create a learning analytics pipeline that transforms student monitoring and observations into actionable insights for teachers as part of reflection and feedback activities, or for dynamic intelligent adaptation of the game itself.

This document is the 2nd in a series of deliverables describing the detailed system design and implementation of prosocial affect fusion and player modelling capabilities within the ProsocialLearn architecture. The 1st version (D3.2) was delivered at M10 (October 2015) and describes a core domain based approach to measuring prosocial behavior along with sensor approaches to measure engagement and emotion. The document now considers the implications of the updated user requirements and conceptual framework for teaching prosociality detailed in D2.1 User Requirements. The primary change is teaching children based on the psychologist’s view of prosocial domains to teaching using the pedagogical approach of Skillstreaming. The skill streaming approach changes the way in-game data is measured and incorporated into the player model but does not change the observation of emotion and engagement. The final version (D3.4) will be delivered at M24 (December 2016) describing the final version of game technology capabilities.

1.2 Scope and Audience of the document

This section scopes the deliverable in relation to the overall ProsocialLearn architecture. The components primarily contribute to key elements of the Prosocial Learning Analytics Pipeline.

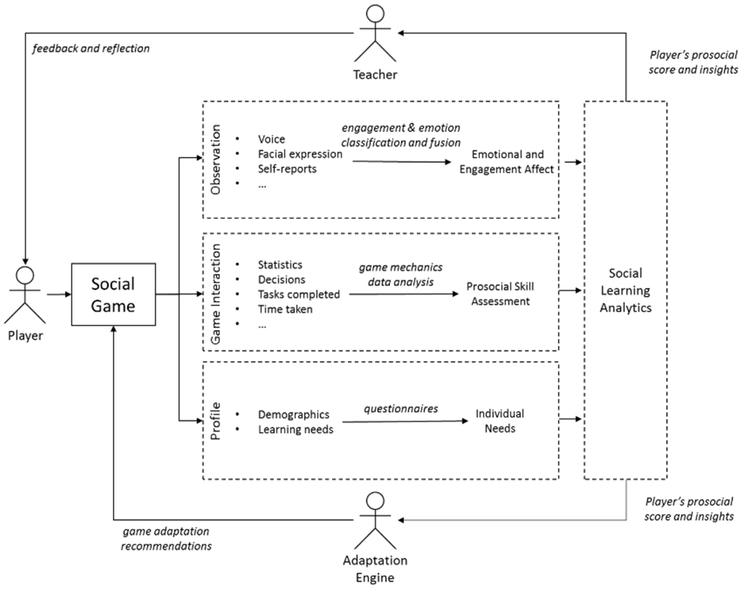

Teacher
feedback and reflection
Player's prosocial score and insights
Observation
Voice
Facial expression
Self-reports
...
engagement & emotion classification and fusion
Emotional and Engagement Affect
Player
Social Game
Game Interaction
Statistics
Decisions
Tasks completed
Time taken
...
game mechanics data analysis
Prosocial Skill Assessment
Social Learning Analytics
Profile
Demographics
Learning needs
questionnaires
Individual Needs
Player's prosocial score and insights
game adaptation recommendations
Adaptation Engine

Figure 2: Overview of the learning analytics platform.

An overview of the entire pipeline is shown in Figure 2. The learning analytics pipeline transforms student monitoring and observations into actionable insights for teachers. The view is that the player modelling informs everything. This is how the platform sees the players (students). Interactions with the platform to update/create the model happens via a skills based approach. At runtime skill is acquired through game interaction monitoring whilst temporal emotional state is observed through multi-modal sensors analysing voice and facial expression. The sensors we use are these and we use them to measure emotion and engagement (link to WP4). To be meaningfully understood we need to fuse the sensors. The skills and the fused sensors are presented in the dashboard along with the information teachers want. The data is stored across multiple games as part of a user profile and combined with off-line questionnaires capturing additional information such as demographics and cultural context.

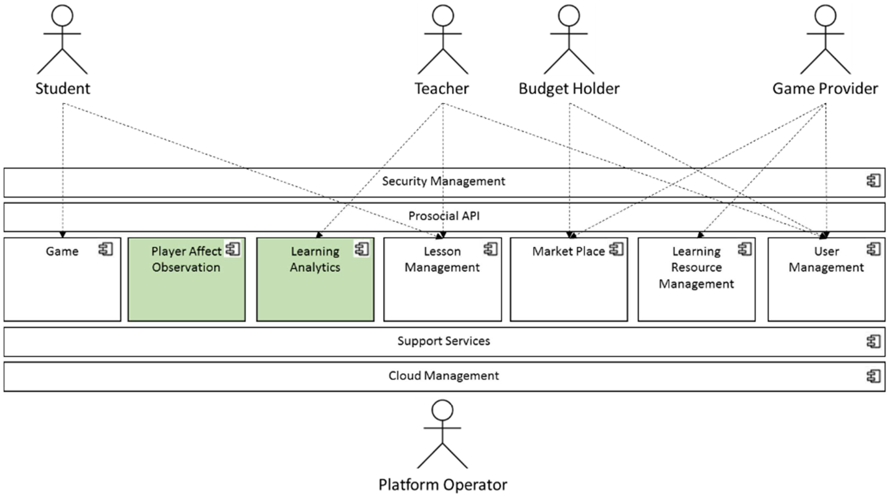


Figure 3: Scope of the ProsocialLearn subsystems covered by this deliverable

A conceptual view of the high level architecture of the ProsocialLearn platform is shown in Figure 3. Figure 4 shows the components of the architecture is scope of this deliverable. The green coloured subsystems are within the scope of this deliverable. Specifically, this deliverable deals with ways to process the incoming data from the prosocial games to augment models of the player's prosocial behavior. In addition we are concerned with the presentation of the derived to Teachers via a dashboard. The player affect observations come via external sensors or specific in-game instrumentation. The data is used to derive measures of the player's emotion and engagement. As these measures come from a number of different sources fusion is employed to provide augmented estimates. These estimates combine the individual results to produce a more precise result with an associated error.

The learning analytics subsystem combines the emotions, engagement along with game events and the player profile to create an estimate of the current "prosocial score" of the player. This score is an instantaneous measure of the state of a player's prosociality. This is stored alongside the player's profile. This can be used to create short- or long-term measures of the player which can be used by teachers to decide if intervention is needed. This in-situ analysis of the data is performed by the prosocial analytics sub-system.

The final component within scope is the student performance dashboard. The objective of this is to provide visualisations of the underlying data to provide insights to the teacher by allowing exploration of the data to understand how best to provide feedback to students. The Student Performance dashboard (in blue) will be reported in the final version of the deliverable.

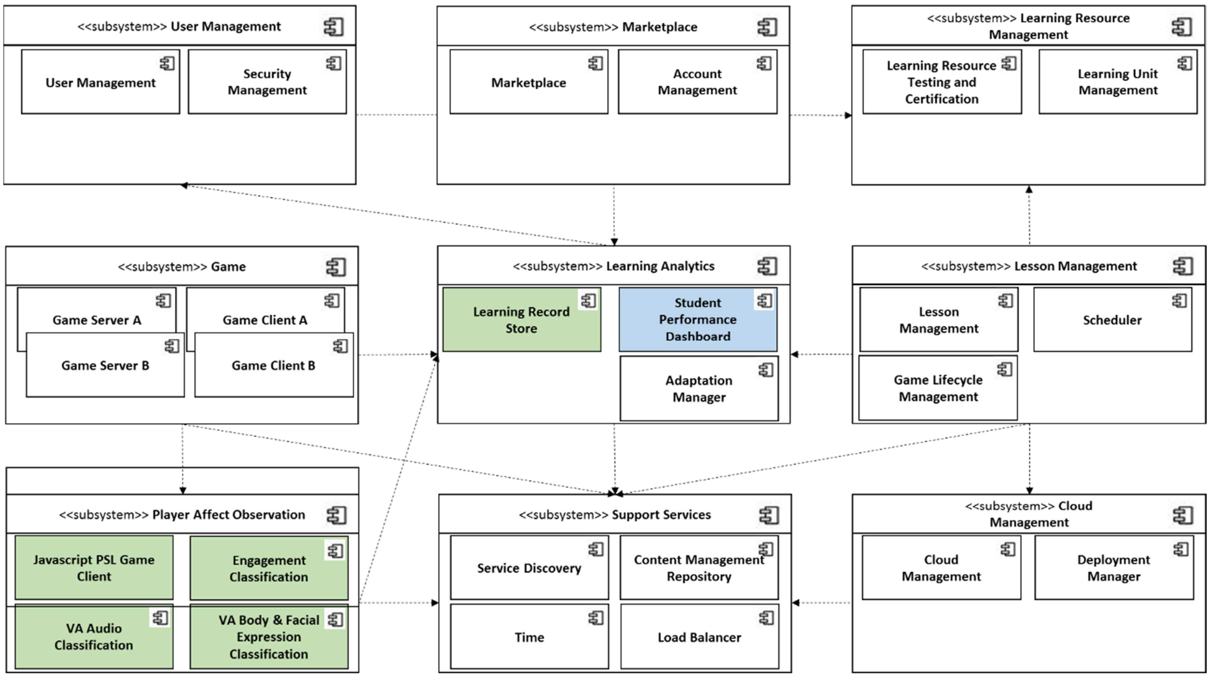

<<subsystem>> User Management
User Management
Security Management
<<subsystem>> Marketplace
Marketplace
Account Management
<<subsystem>> Learning Resource Management
Learning Resource Testing and Certification
Learning Unit Management
<<subsystem>> Game
Game Server A
Game Client A
Game Server B
Game Client B
<<subsystem>> Learning Analytics
Learning Record Store
Student Performance Dashboard
Adaptation Manager
<<subsystem>> Lesson Management
Lesson Management
Scheduler
Game Lifecycle Management
<<subsystem>> Player Affect Observation
Javascript PSL Game Client
Engagement Classification
VA Audio Classification
VA Body & Facial Expression Classification
<<subsystem>> Support Services
Service Discovery
Content Management Repository
Time
Load Balancer
<<subsystem>> Cloud Management
Cloud Management
Deployment Manager

Figure 4: Scope of the Prosocial Learn components covered by this deliverable

1.3 Structure of the document

The document is split into the following high level sections.

Section 2 covers how the player data is modelled

Section 3 describes how the prosocial skills are measured

Section 4 presents the observations present in the ProsocialLearn platform

Section 5 illustrates the approach used to multimodal fusion

2 Player Modelling

2.1 The role of Player Modelling in ProsocialLearn

The term 'player' and 'student' is often used interchangeably in PSL and refers to the child that plays prosocial games as part of a learning experience enabled by a learning provider. These two terms are used as they are required in context: 'student' is used when referring to the child in the wider learning context (a school class, for example) and 'player' is used when describing activities related to the child whilst at play with PSL games. In modelling PSL 'students' or 'players' we base our approach on well-established foundations developed by the user modelling community. User modelling is a method of structuring and updating data related to end-users of a system with the view to using it as input to an adaptive process that modifies system behaviour in specific ways that improve interaction outcomes. There are a variety of stakeholders within the project that are identified as users of the PSL platform – these roles have been identified in D2.4 System Requirements and Architecture. However, our scope here is limited to the interests of those users who directly take part in prosocial games. Readers should note that other end-users of the PSL platform have an interest in the player/student model data (namely teachers, parents and experimental psychologists) but they themselves are not the primary focus of this model.

Our definition of the PSL player and student model is driven by the requirements generated during the early phases of the project's development and guided by user modelling methods found in the research literature. When used in application design, user models are typically representative of a relatively narrow view on users working in a specific problem domain within a particular context (Clemmensen 2004). For the sake of brevity, much of the supporting literature behind this work is given in appendix B. In the case of PSL our specific focus on the user relates to their prosocial behaviours and emotional and engagement responses during game play. In the sections that follow, we present the principal data elements of the model; describe where such data is generated in the PSL platform; how it is specified and stored; and provide an overview of the means by which the data is subsequently accessed by other components of the system.

2.2 Model principal elements

We consider two orthogonal data sets in the representation of the user: the *student profile* and the *player history;* see the figure below.

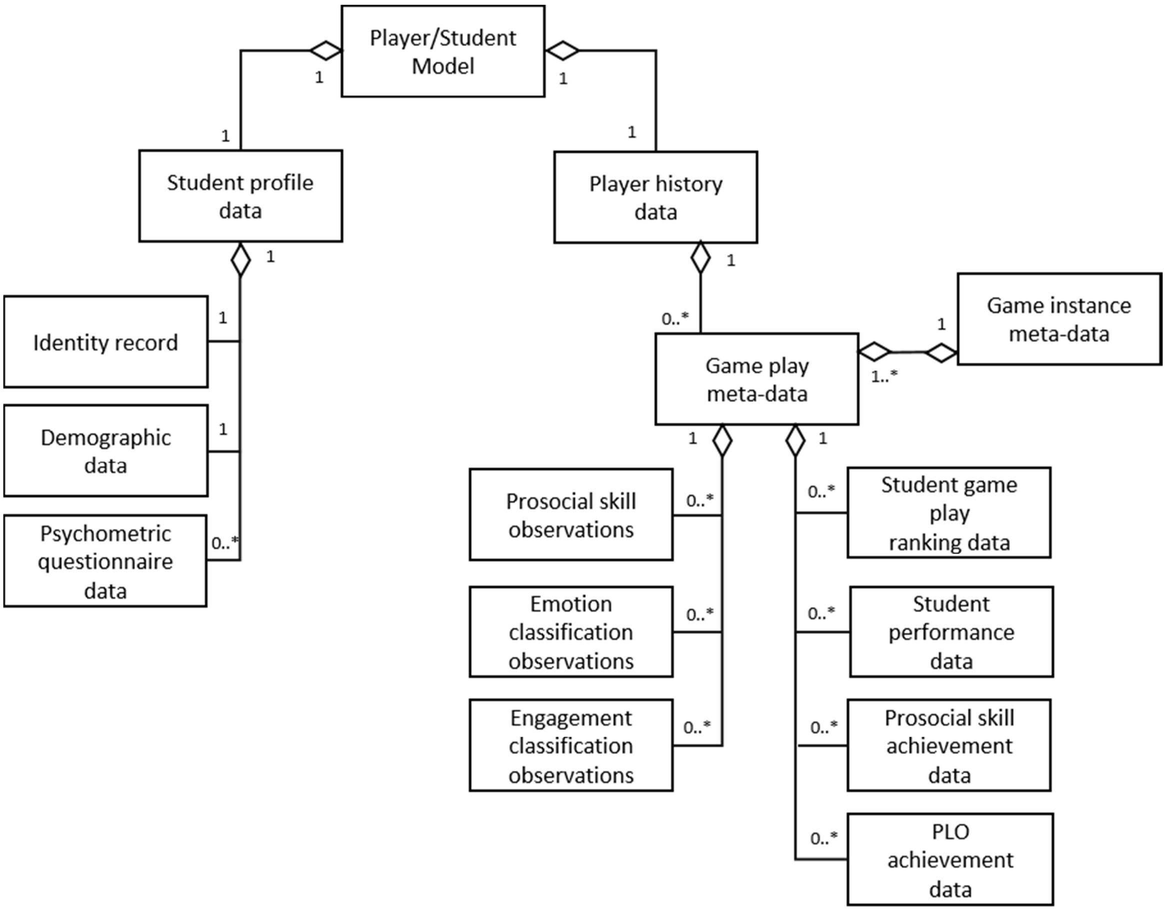


Figure 5: Player model relational overview

The former characterizes aspects of the student's identity and background information (in the context of the relationship to the learning provider) and psychological characteristics. In the latter case, the player history data records prosocial, emotional and engagement behaviors as well learning outcomes (and related data) captured during game play.

2.3 Student profile data

The student profile data is anticipated to serve two main purposes. First, to provide sufficient identity and contextual information to allow the teacher to manage game classes effectively using the PSL Teacher Dashboard as part of the Lesson Management subsystem. Second, to provide references to associated experimental data (principally in the form of questionnaire responses) that will assist in the project's experimental analysis of game play results. This data is intended to be used to support the scientific analysis carried out in the project as a means of identifying and controlling for variation in participant characteristics that may impact the outcome of an experiment. The selection, administration and collection of questionnaire (or test) data will be particular to the experiment being carried out; considered private and sensitive; and is expected to be managed separately and securely. For this reason, only references to uniquely identified response data sets should be made in the student profile. Examples of the characteristics of experimental participants that could be captured in experimental case studies include representations of social exclusion; antisocial behaviour; academic performance; and personality traits – see D2.5 Evaluation Strategy and Protocols for further information.

2.4 Player history data

The player history elements of the model encapsulate observations made directly or indirectly (via a classification process) of players' behaviour during game play. Historical data will be used for three purposes: (i) as input to the PSL game adaptation algorithm, (ii) as information to be used by teachers in the assessment of students' progress and (iii) as data for experimental analysis. The features and application of the player history data to game adaptation will be evaluated in game studies over the course of the project; modifications and extensions to the model (such as the inclusion of social graph data) may be applied in future versions. Data sources will vary depending on the type of observation that will be used; this is described further in section 2.7.

Players will be associated with data related to game instances they have played, including:

- Game generic info (such as game time and outcomes)
- Anonymized references to other players in the game
- Game interactions associated with them over time
- Classifications of their emotion over time
- Classifications of their engagement over time
- Acquisitions of prosocial skills/learning objectives achieved over time

A subset of the game information model (see D2.4) provides the basis for structuring game player history data. In the right half of Figure 5 those elements used to encapsulate player behaviors observed during game-play and their subsequent evaluation are presented. Over the course of a single game instance, each player has related 'game play meta-data' – this represents the collection of observations and evaluations for that player. Prosocial game interactions; emotion and engagement classifications will be transcribed using the 'Prosocial Learning Specification' (PLS) language (see section 2.5.2) developed by the project for the purpose of representing and capturing this data. Evaluations of game play, based on this data, will also be recorded in a similar way; this provides us with a rich data set that uniquely describes that student's learning experience and outcomes (see section 3.2).

2.5 Model presentation

2.5.1 Student profile data presentation

As already indicated, student profile data is to be stored and accessed securely by a limited subset of actors within the PSL architecture. This data will be stored in a relational database and managed using a conventional schema (see Figure 6).

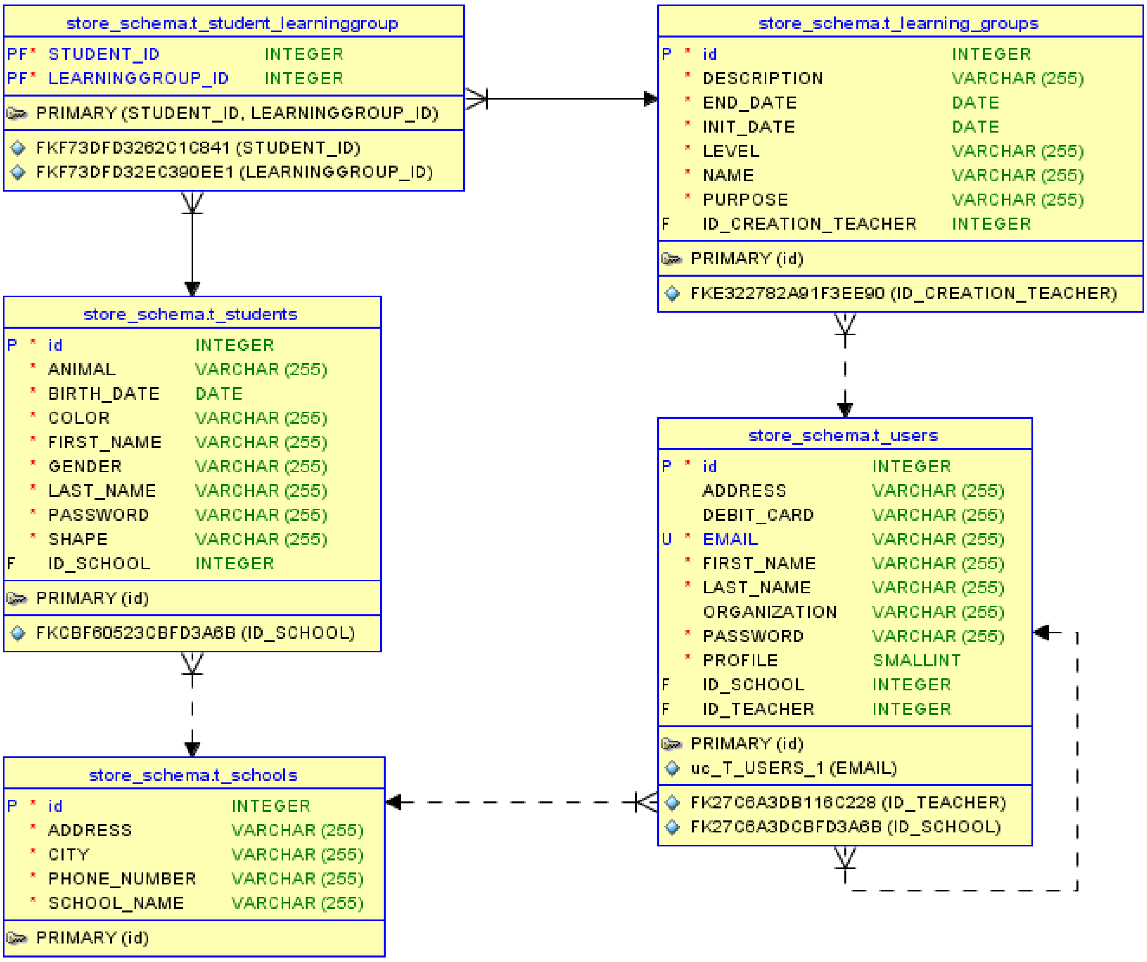


Figure 6: Core student profile schema

2.5.2 Player history data representation

As the project has progressed we have continued to evaluate our design strategy for representing player history data (in-line with the developmental processes described by (Cocea & Magoulas 2015)). As a result, we have iterated our approach toward representing player history by adopting a format that is a) refactored to toward the skills based approach now using in PSL and b) widely adopted by technologists and game companies working in the field of education gaming and c) better aligned with the ontology, graph based approach used in the user-modelling community. To this end we have chosen to adopt the 'Experience API' or 'xAPI' specification[1] as a foundation for defining the PLS language that will be used to encapsulate player history data.

The 'xAPI' is a specification that "makes it possible to collect data about the wide range of experiences a person has (online and offline). This API captures data in a consistent format about a person or group's activities from many technologies" [see API introduction]. At the time of writing, the official release is in version 1.02; the specification provides a description of (learning) experiences using a simple statements centered on actors enacting actions (as verbs) on objects – a range of

[1] See https://github.com/adlnet/xAPI-Spec/tree/1.0.2

additional meta-data can also be attached to these statements. Such statements can be sent from a wide variety of technology contexts (see the technical resources here) and are aggregated in a Learning Record Store (LRS). The role of LRS is primarily as a data management service – it provides a RESTful HTTPS endpoint that is used to receive xAPI statements and also provide query facilities to generate data for analytical purposes.

From the stand-point of understanding and communicating (prosocial) in-game interactions and player observations (related to emotion and engagement) our technical analysis has lead us to adopt an xAPI based representation for a number of important reasons. Specifically these are:

- xAPI language provides a powerful and easily 'tailorable' language that is specifically designed to describe learning based interactions between people and software systems.
- xAPI provides a well-defined set of queries that significantly expedites the creation of data analytics related (but not limited) to learning performance.
- xAPI is a well-established standard that is supported by Learning Record Store (LRS) implementations from both open source communities and commercial enterprise.
- Game developers within the educational market already understand and use xAPI.

2.6 Prosocial Learning Specification statements

This section assumes the reader has some understanding of the underlying xAPI specification and its high-level structures[2]. The prosocial learning specification closely follows the 'actor-verb-object [+context]' methodology set out by the xAPI standard and reflects those aspects of the relevant architectural and pedagogical models set out in D2.4 and D4.1 respectively. To this end, the PLS is built around:

- *Actors* (e.g. Teacher, Student, Learning Group)
- *Verbs* (e.g. 'Created', 'Started', 'Stopped', 'Helped', 'Shared' etc)
- *Activities* (e.g. Lesson, Game instance, Game situation)

Prosocial Learning Specification statements will be generated both by the PSL platform and the games themselves during runtime. In Figure 7, we present high-level information model overview, syntactical examples of statements from both of these contexts.

Ahead of actual game-play, it is anticipated that teachers will use the PSL platform to define lessons in which groups of students play particular games. Essential elements of the results of this planning procedure will be recorded by the Learning Record Store for the purposes of later making queries on game-play outcomes more easily accessible. So for example, PLS statements will describe a teacher's activities in creating new lessons and assigning game instances to learning groups. Later, during the course of the planned lesson, actions related to 'live' game instances (such as emotions expressed or game adaptations) are also registered within the LRS via PLS statements.

By far the most frequently generated PLS statements will be those that relate to player behaviors during game play. These framed between the beginning and ending of particular 'game situations' (a role or scenario particular for a player); the identification of a particular skill that has been exercised to some extent by a player (for example, a sharing interaction); and the presence of certain emotional and engagement responses as observed by the PSL classifiers. Each of these kinds of statements will be sent by the game to the LRS.

[2] Readers unfamiliar with this formalism and related technology are directed to: https://experienceapi.com/overview/

Contextual information plays an important role in xAPI statement construction and the PLS make use of this data where appropriate (but avoids redundancy where possible). Experience API based statements often require context; in the figure above a game instance is contextualized with references to a) other PLS activities relating to the lesson and/or b) inline context data identifying other associated entities (here this would be the game learning group and PLO related information).

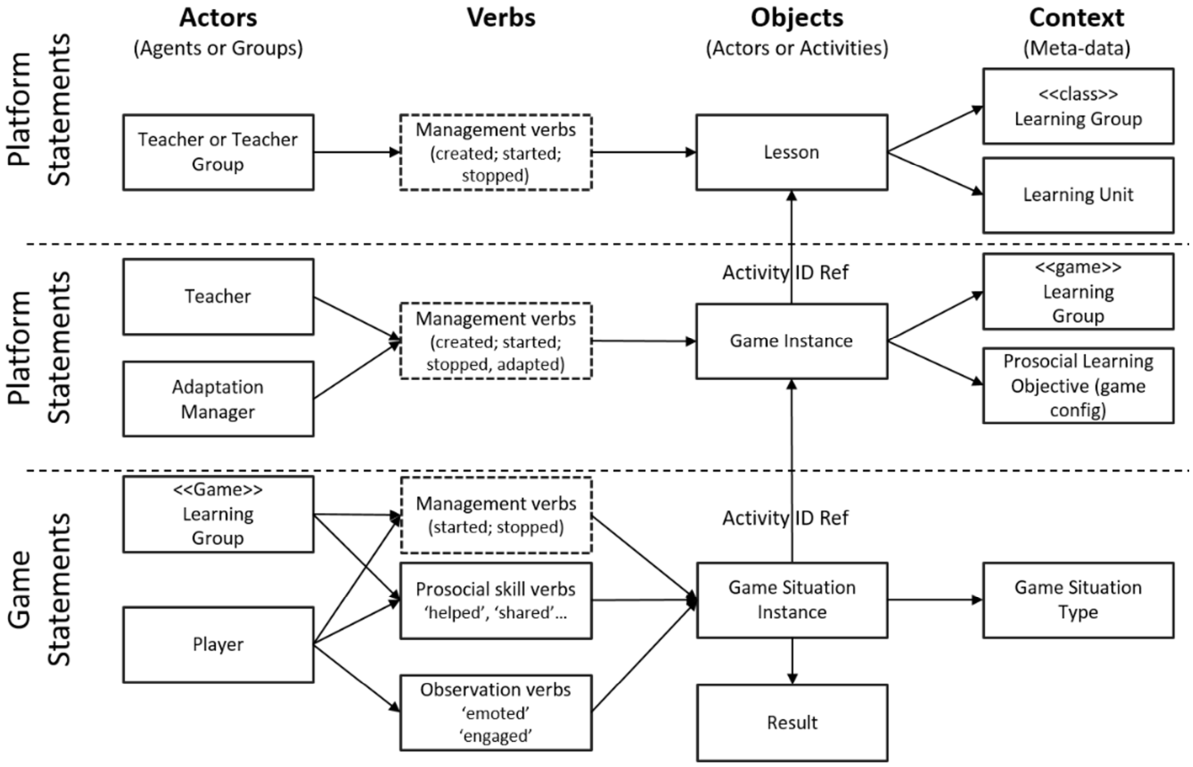


Figure 7: PLS information model overview

PLS naming methodology

In order for PLS statements to be constructed, we must first have a consistent and xAPI compliant method for referring to teachers, students, game instances and so on that are managed by other parts of the PSL platform. This naming format is set out below for each of the main elements identified in the xAPI based PLS formalism.

Table 1: PLS actor identifier examples

Actor	xAPI type/ID	Example

Learning Provider	Group/ IFI:account	`{ "objectType": "Group",`
		`"account": { "homePage": "http://prosociallearn.eu/",`
		`"name": ‶InstitutoVirgoCarmeli‶ }}`

School Admin	Agent/ IFI:mbox	`"mailto:admin@virgocarmeli.it"`

Teacher	Agent/ IFI:mbox	"mailto:montessori@virgocarmeli.it"
Learning Group	Group/ IFI:account	{ "objectType": "Group", "account": { "homePage": "homePage": "http://prosociallearn.eu/lgids/", "name": “438fde14-e8c6-4847-8815-e7453c2253f4" }}
Student	Agent/ IFI:account	{ "objectType": "Agent", "account": { "homePage": "http://prosociallearn.eu/pids/", "name": “1901e039-7b26-41cb-98da-6381240c8ee1" }}

Actors are identified by using either xAPI ‘accounts’ or FOAF[3] mbox identifiers. It is assumed that the Learning Provider can be uniquely identified by its real-world institutional name. Staff at the Learning Provider are expected to have an email account that will identify them. Finally, UUIDs are used to anonymously identify students and their learning groups (personally identifiable student information and related data is stored elsewhere in the PSL platform); these UUIDs are appended as ‘names’ to the appropriate IRI [homepage value] in the account field).

Table 2: PLS verb identifier examples

Verb	xAPI PLS Verb examples
Created	{ "id": "http://prosociallearn.eu/plsxapi/verbs/created","display": { "en-US": “created" } }
Started	{ "id": "http://prosociallearn.eu/plsxapi/verbs/started","display": { "en-US": “started" } }
Stopped	{ "id": "http://prosociallearn.eu/plsxapi/verbs/stopped","display": { "en-US": “stopped" } }
Adapted	{ "id": "http://prosociallearn.eu/plsxapi/verbs/adapted","display": { "en-US": “adapted" } }
Emoted	{ "id": "http://prosociallearn.eu/plsxapi/verbs/emoted","display": { "en-US": “emoted" } }
Engaged	{ "id": "http://prosociallearn.eu/plsxapi/verbs/engaged","display": { "en-US": “engaged" } }
Helped	{ "id": "http://prosociallearn.eu/plsxapi/verbs/helped","display": { "en-US": “helped" } }

All PLS verbs have a pre-defined IRI namespace (http://prosociallearn.eu/plsxapi/verbs/) and are considered a fixed part of the PLS syntax. In the table above we show some examples from this namespace that refer to platform and game related interactions described in Figure 7

[3] http://xmlns.com/foaf/spec/

Table 3: PLS Object identifier (activities) examples

Object	xAPI type/ID	Example (identifier)

Lesson	Activity /PLS IRI	`"object": {`
		`"id": "http://prosociallearn.eu/aiids/45b6fe6f-3865-4b40-a662-c8a742cfdfb6",`
		`"definition": { "name": { "en-US": "Monday's lesson" } },`
		`"objectType": "Activity" }`

Game instance	Activity /PLS IRI	`"object‟ : {`
		`"id": "http://prosociallearn.eu/aiids/27bee60c-e1d1-4fb5-b3cc-b32eeb7bd507",`
		`"objectType": "Activity‟ }`

Game situation	Activity /PLS IRI	`"object‟ : {`
		`"id": "http://prosociallearn.eu/aiids/ebb7661d-43cc-45be-84fa-81b32cfb0c93",`
		`"objectType": "Activity‟ }`

Finally, xAPI 'objects' form the target of the statement. In xAPI objects may resolve to actors, activities or other statements – these are be used appropriately in PLS, depending on the nature of the statement. In the examples provided above, we show how instances of activities (which must be uniquely identified) are named. All instances of activities are identified by UUIDs which map to the same UUID values used in other parts of the PSL platform. For example, a new lesson instance (created by a teacher using via Lesson Management subsystem) will have a UUID internally associated. The lesson management component of the portal will communicate this UUID to the LRS in a statement about the creation of a lesson. Later, when a game instance is prepared for the lesson, its UUID will also be communicated to the LRS, along with a contextual reference to the lesson within which it will be played. In a similar fashion, the game server running the game instance will report the creation of specific game situations (these are activities specific to a player) and use the related game identifier as contextual information. All activity instance identifiers have a pre-defined IRI namespace (`http://prosociallearn.eu/aiids/`) followed by their UUID.

Example PLS statements

In the table below we provide a PLS statement example in which a teacher creates a game instance for a lesson.

	xAPI Type	Identified by	Example
Actor	Agent	mbox IFI	"mailto:montessori@virgocarmeli.it "
Verb	Verb	PLS IRI	"http://prosociallearn.eu/plsxapi/verbs/created"
Object	Activity	Activity	"id": "http://prosociallearn.eu/aiids/aa7b5c72-914e-4733-b528-1a88b5b139b7",

		instance IRI	"definition": { "name": { "en-US": "Game 1" }, "type": "http://prosociallearn.eu/plsxapi/activityTypes/gameInstance”
Context	Group	Account IFI	"team": { "homePage": "http://prosociallearn.eu/lgids/","name": "438fde14-e8c6-4847-8815-e7453c2253f4" }

Table 4: PLS statement outline (teacher registration)

Here the teacher is an actor uniquely identified by an email address (montessori @virgocarmeli.it). The verb describing the specific activity is also uniquely identified by a verb taken from the PLS verb name space (in this case, ‘created’). The target of the activity (the object) is in this case an activity instance which maps to the UUID that identifies a prepared game instance on a game server. Additional contextual information provided in the context elements of the statement provides us with meta-data that supports PLS querying services. We provide one such example of this above, the ‘team’, refers to the (game) learning group identifier associated with the game instance. Other methods of reference, such as a pointer to another PLS statement specifying a learning group are also possible. This data is encapsulated in JSON (see below) and sent to the LRS via a secured HTTPS POST statement.

```
{
    "actor": {
        "mbox": "mailto:montessori@virgocarmeli.it",
        "objectType": "Agent"
    },
    "verb": {
        "id": "http://prosociallearn.eu/plsxapi/verbs/created",
        "display": { "en-US": "initialized" }
    },
    "object": {
        "id": "http://prosociallearn.eu/aiids/aa7b5c72-914e-4733-b528-1a88b5b139b7",
        "definition": { "name": { "en-US": "Game 1" },
            "type": "http://prosociallearn.eu/plsxapi/activityTypes/gameInstance"
        },
        "objectType": "Activity"
    },
    "context": {
        "team": {
            "account": {
                "name": "438fde14-e8c6-4847-8815-e7453c2253f4",
                "homePage": "http://prosociallearn.eu/lgids/"
            },
        },
        "revision": "1.0",
        "platform": "PSL"
        }
}
```

Table 5: PLS JSON encapsulation of a teacher creating a game instance

Example PLS emotion classification statement

By way of an example of player history data, in the table below we illustrate how an emotional classification (generated by the fusion service) is described using the PLS schema:

	xAPI Type	Identified by	Example
Actor	Agent	Account IFI	{ "objectType": "Agent", "account": { "homePage": "http://prosociallearn.eu/pids/", "name": "1901e039-7b26-41cb-98da-6381240c8ee1" }}
Verb	Verb	PLS IRI	"http://prosociallearn.eu/plsxapi/verbs/emoted"
Object	Activity	Game situation IRI	"id": "http://prosociallearn.eu/aiids/f2431850-e0c3-4a6c-955b-ffc1086cb1b4"

Result	Extension	JSON extension element
		"v" for valence value
		"a" for arousal value
			{
			"vaObserver" : "FusionService",
			"observation" : { "v" : 1.0, "utc" : "2016-07-14T14:41:54.319Z" } }

In this example, we refer to the player being observed (via means of a xAPI account identifier) in the actor slot of the xAPI statement. We note that it is typically a software process that is making the assertion about the emotion (not the player) but here the identity of the software process is a second-class data element. The primary use of this data is to understand the emotional responses of a player. The object of the statement is the unique identity of the game situation instance being played at the time (this ID is maintained by the game server). In the result extension element of the xAPI statement we record a valence/arousal time-stamped measurement. Finally, the type observer making the assertion relating to player emotion is provided (this differentiates machine generated assertions from those made by the game player herself using the Emotion Self Report client, for example).

Technical application of the Prosocial Learning Specification

PLS statements will be used extensively within and without the ProsocialLearn platform. Internally, PLS statements will be transmitted to record prosocial interactions; emotions and engagement measurements (as described above) and also used to record some aspects of the learning environment supported by the PSL platform (such as describing lesson activity within schools). The PLS is under development and is an evolving specification that will be iteratively updated and released to technical partners over the course of the project. Our early release of the PLS currently covers the following game play related elements:

- Registering a learning provider with the PSL platform provider
- Closing a learning provider's PSL account
- Managing teachers within a PSL account
- Managing students within a PSL account

- Managing learning groups within a PSL account
- Representation of valence/arousal measurements of game players

This specification will provide technical partners with the syntactic information required to send and retrieve PLS-xAPI based statements to and from the LRS. In addition to this, a lightweight 'wrapping' API is under development that will offer rapid methods for generating and sending PLS statements programmatically.

2.7 Model acquisition and persistence

ProsocialLearn user model data will be captured from a number of different sources and modalities. Student profile data is expected to be acquired from parents and children using conventional forms of questionnaire administration, the individual responses of which will be managed separately. This data is expected to be used (in anonymous form) for experiment analysis purposes and will not be directly accessible by other parts of the fusion platform.

Player modelling data on the other hand is captured from sources directly present during each game played; this data is much more dynamic and is generated through a series of processing pipelines – see the figure below.

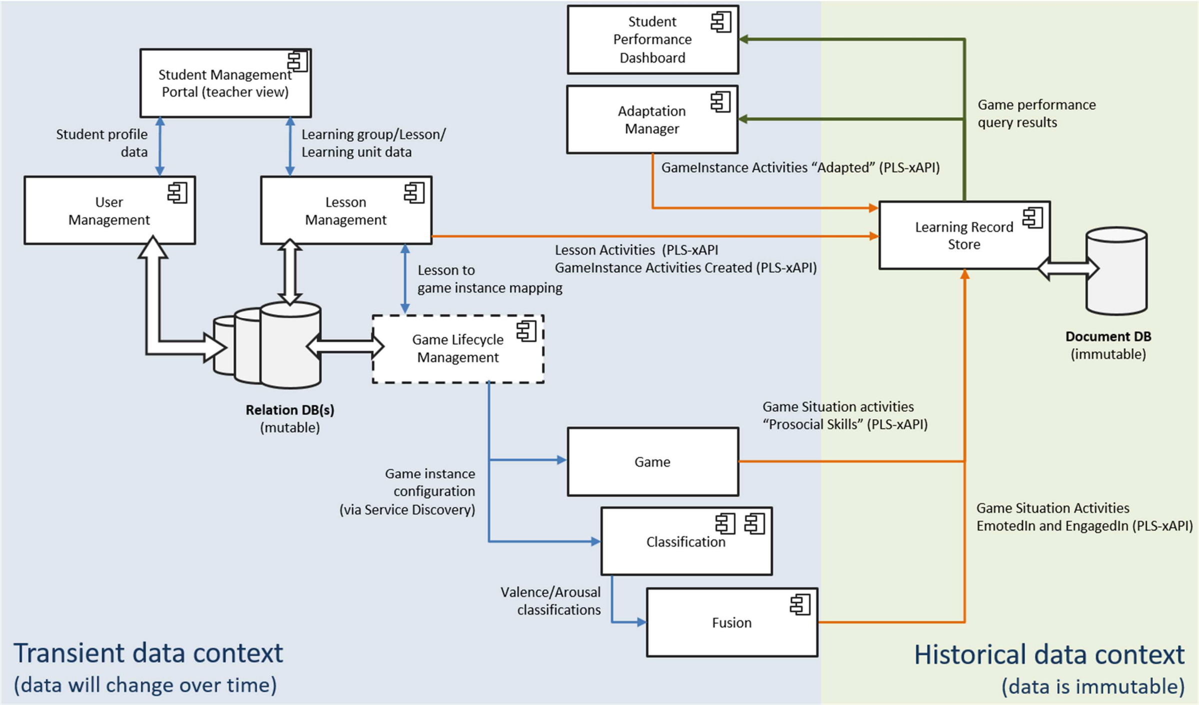

Student Performance Dashboard
Student Management Portal (teacher view)
Adaptation Manager
Game performance query results
Student profile data
Learning group/Lesson/ Learning unit data
GameInstance Activities "Adapted" (PLS-xAPI)
User Management
Lesson Management
Learning Record Store
Lesson Activities (PLS-xAPI
GameInstance Activities Created (PLS-xAPI)
Lesson to game instance mapping
Document DB (immutable)
Game Lifecycle Management
Relation DB(s) (mutable)
Game Situation activities "Prosocial Skills" (PLS-xAPI)
Game
Game instance configuration (via Service Discovery)
Game Situation Activities EmotedIn and EngagedIn (PLS-xAPI)
Classification
Valence/Arousal classifications
Fusion
Transient data context (data will change over time)
Historical data context (data is immutable)

Figure 8: Player modelling data generation, acquisition, persistence and storage

Indications of emotional and engagement responses from the players are generated through observations using a number of sampling techniques that feed feature detection and classification processes (see section 4 for a detailed description of these methods). Once a decision on the emotion and/or engagement of a player has been made by the fusion service it is encoded using the PLS language and sent to the Learning Record Store (LRS) for persistence. Down-stream services that depend on this data (including Learning Analytics and the Adaptation Service) use PLS based queries to retrieve data used to assess student performance for a number of purposes (for example, the Adaptation Service would make recommendations to game servers based on these results).

Player modelling data persistence

Our adoption of an xAPI based representation of player model data allow us to expedite the development of the PSL platform by considering existing and proven LRS implementations on which we can build specific support for prosociality in games. In doing so, this allows us to:

- Develop the PLS language, built on top of the proven xAPI formalism, allowing us to focus just on relevant game interactions, skills description; and emotion and engagement classification outcomes.
- Use a proven LRS implementation that has data management capabilities 'out-of-the-box' allowing us to focus data analytics and end-user needs (such as the UI for teachers).
- Rapidly integrate with an already implemented xAPI web based end-point, providing developers with a mature and documented to work with (both internal to our PSL platform, or external to 3rd party game developers wishing to use it).

Our recent analysis of currently available open source LRS implementations (including RAGE analytics, ADL LRS, OpenLRS, lxHive and Learning Locker) considered each offering along a number of criteria including technical capability; deployment and security options; technical maturity, adoption and liveness; and dashboard/data analytics offerings. We reached the conclusion that the most suitable LRS implementation was Learning Locker[4], which was rated highest overall in our analysis.

2.8 Qualified access to student profile/player history data

Access to both student profile and player history data needs to be secured for both user-facing interactive applications as well as middleware services running on the PSL platform. The scope of the 'data view', as well as the means by which it is accessed, will depend on the platform components wishing to gain access to the data. Enabling qualified access to the data will require that system components be able to 'sign-on' using identities that afford them varying degrees of access to data related to individuals or groups. For example, let us compare views of the profile and player history data as they relate to the requirements of teachers, parents and experimental psychologists. The teacher's view of student model data will support game planning and student assessment and may require access to multiple instances of profile/historical information linked directly with student identities. Each child's parent or guardian is expected to have access to the same information, but scoped only to their children. An experimenter's view of student model data is typically framed within the perspective of comparing groups of (anonymized) data sets related to game instances with the aim of finding correlations or significant differences. In Table 6 we explore use-cases in which varying access to profile and history data are outlined.

Use-case	**End-user/service**	**Scope**	**Personal data required?**	**Read access**	**Write access**
Student profile questionnaire responses (by parent or teacher)	Parent or Teacher	Individual	Y	Personal identity data	Questionnaire data

[4] https://learninglocker.net/

	User Management Service				
Game lesson planning using the Student Performance (SP) dashboard	Teacher SP dashboard	Group	Y	Game history data	None
Configuration of game settings before start	Game server	Group	N	Game history data	
Recording of prosocial interactions: game instance usage and in-game achievements	Game server	Individual	N	None	Game history data
Review game progress at game-time	Teacher SLA dashboard	Individual & Group	Y	Game history data	None
Evaluate game outcomes; record notes on lesson.	Teacher SP dashboard	Individual & Group	Y	Game history data	Game history data
Assembly of game data for experimental analysis	Psychologist SP dashboard	Group	N	Questionnaire data Game history data	None

Table 6: Use cases for access to the user model data

It is clear that well defined roles and access policies should be defined such that services interfacing the student and player model data, which may include private data, only do so when it is necessary and only with the correct authorization. The security implementation for access control is described in D2.4

3 Measuring Prosocial Skills

Teaching social skills through digital games in school environments requires a robust scientific approach to maximise the potential positive benefits to students, to increase acceptance of novel game-based learning by teachers and to provide game designers a methodology for creating games that are effective. Evidence-based research indicates that successful learning of social skills requires well-designed classroom-based programs that target the range of prosocial competencies, provide opportunity to practice, and offer multi-year programming. Prosociality is an abstract concept that is conceptualised, investigated and applied within the disciplines of psychology and pedagogy. Developmental psychology has shown that prosociality can be understood using domains (Eisenberg & Mussen, 1989) such as empathy, trust, fairness, generosity and cooperation. Although the domains are useful in explaining prosocial concepts that children need in order to be successful learners and be socially included, the concepts are complex social constructs that are difficult to define, measure and incorporate into measurable learning objectives. From a pedagogical perspective the Collaborative for Academic, Social, and Emotional Learning (CASAL) (Zins et al. 2004; Bridgeland et al. 2013) and Skillstreaming (McGinnis & Goldstein 1997) offer practitioners systematic approaches to teaching social skills. The CASEL framework offers five social and emotional learning competencies: self-awareness, self-management, social awareness, relationship skills and responsible decision making whereas Skillstreaming identifies 60 skills deemed necessary for prosociality that are lacking in students. Skillstreaming focuses on sequence of learning strategies; instruction/description, modelling, role-playing, performance feedback and generalisation (trying the skill in different context). Both skill-deficit, where the child lacks the know-how about a behavioural skill, and performance-deficit, where the child is aware of the correct behaviour but fails to reproduce it in the correct circumstance, are addressed through this technique. For example the child may have the know-how on how to carry out a skill but because of lack of positive reinforcement, or lack of confidence the child does not perform the skill in the appropriate setting. PSL adopts a skills based-approach to learning social skills. We have identified an initial set of 40 skills within three classes: skills for friendship, skills for feelings, and skills for collaboration. The skills were selected considering their applicability to and benefit from digital game-based learning, for example, the skill can be measured through sensor observation and monitoring tools. The skills are also of different difficulties and can be incrementally learnt to progress students through levels of prosociality. For example, identifying feelings is necessary to be able to showing concern for other's feelings or dealing with angry feelings. Each game can be used to learn one or more skills depending on the nature of the game situations, decisions and mechanics.

Classification	Prosocial Skills
Skills for friendship	Communicating with others, Using Nice Talk, Introducing Self to Others, Introducing Others, Joining in a Conversation, Joining a Play Group, Sharing About Oneself, Sharing Your Things With Others, Learning About Others, Being an Active Listener, Giving Compliments, Receiving Compliments, Respecting Others, Respect for Others' Personal Space, Not Interrupting Others
Skills for feelings	Self-Control, Identifying Feelings and Emotions, Expressing Feelings and Emotions, Understanding Social Cues, Showing Concern for Others' Feelings, Dealing With Stress, Dealing With Anxiety, Dealing with your angry feelings, Dealing With Another

	Person's Angry Feelings, Dealing With Rejection, Dealing With Being Left Out, Dealing With Boredom
Skills for collaboration	Setting Goals and Obtaining them, Solving everyday problems, Solving a Problem as a Group, Following directions, Paying Attention, Staying on Task, Working Independently, Cooperation, Taking Turns, Being a good sport, Being Patient, Being assertive, Saying No, Accepting No, Asking for Help, Helping Others

Table 7: Initial set of identified skills

3.1 Measuring skills

For each of the skills described in Table 6 work is being carried out to connect the skill with the specific measurements that can be carried out in the game. This mapping is mainly based on the results of in-game mechanics, however, the role of sensors in the measurement of skills will be investigated as well. This work is still ongong and will be reported in the final version of this deliverable (D3.4).

3.2 Monitoring skills

The monitoring of prosocial skills ultimately begins with a particular game developed for the ProsocialLearn platform; each game will have been specifically designed and implemented to exercise one or more particular prosocial skills in children. The PLS language is being developed to integrate descriptions of prosocial interactions (such as instances of cooperation or sharing, as enacted by specific game mechanics) with emotion and engagement responses.

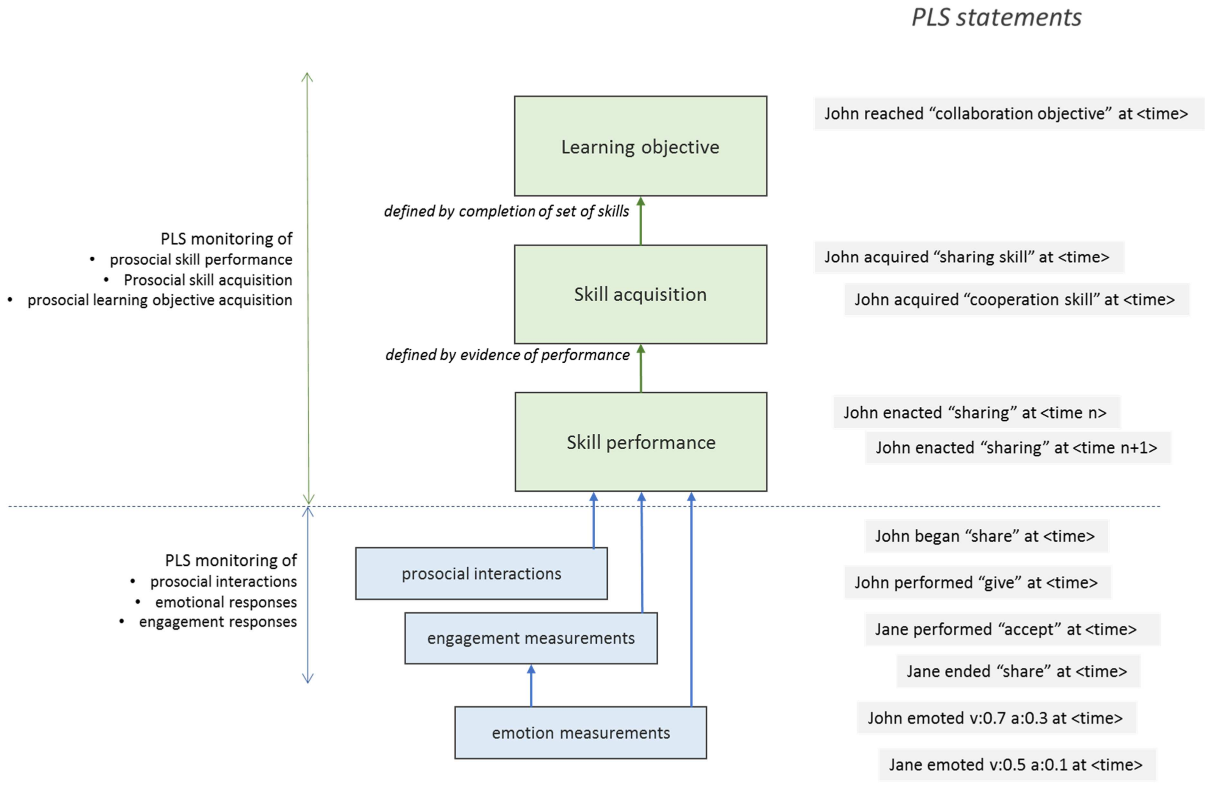

PLS statements
Learning objective
defined by completion of set of skills
Skill acquisition
defined by evidence of performance
Skill performance
prosocial interactions
engagement measurements
emotion measurements
PLS monitoring of
• prosocial skill performance
• Prosocial skill acquisition
• prosocial learning objective acquisition
PLS monitoring of
• prosocial interactions
• emotional responses
• engagement responses
John reached "collaboration objective" at <time>
John acquired "sharing skill" at <time>
John acquired "cooperation skill" at <time>
John enacted "sharing" at <time n>
John enacted "sharing" at <time n+1>
John began "share" at <time>
John performed "give" at <time>
Jane performed "accept" at <time>
Jane ended "share" at <time>
John emoted v:0.7 a:0.3 at <time>
Jane emoted v:0.5 a:0.1 at <time>

Figure 9: Prosocial skills monitoring process

Evidence of these behaviors provides the basis for making further assertions about a student's ability to acquire particular prosocial skills and, ultimately, their completion of prosocial learning objectives. In Figure 9 we illustrate this process as an aggregation of evidence of behaviors as recorded using the PLS in the Learning Record Store for a particular game. In this simple example, John and Jane are playing a game in which prosocial interactions related to sharing (giving and receiving) as well as emotional responses are recorded. At game time, this evidence is aggregated and when certain interactions are grouped in some form, a certain skill can be said to be 'enacted' (this is recorded in the LRS). A skill is said to be acquired here once sufficient evidence of skill performance is identified (this is defined by criteria set by the lesson plan and associated game configurations). Finally, once the requisite skill acquisitions have completed, we say that the student has realized some or all parts of a particular prosocial learning objective.

4 Observations

This section discusses the various observations that are made in the ProsocialLearn platform. Specifically, we describe details of the various input modalities.

4.1 Voice

In D3.1 the classifier was trained using a standard dataset (FAU-AEC[5]). This corpus is a pre-prepared to make the analysis easy. Specifically, the audio stream was cut into small chunks corresponding to a word or a group of words with emotional content. This made training the classifiers easy however for fusion we will have to use continuous audio which is aligned with the other modalities. The alignment process is discussed in section 5.1. This will discuss how to move from discrete to continuous classifiers.

The previous analysis started with pre-emphasis filtering. This stage is kept as it serves a useful purpose in removing noise and flattening the spectrum. The specific implementation is performed via a finite impulse response filter. The signal is then split into overlapping windows. The window serves to split into short term sequences which can be subsequently analysed. A diagram showing the windowing is given in Figure 10. The specifically overlap and offset are tuneable. However, for a 16kHZ audio signal a window size of 512 samples corresponds to approximately 3ms which is similar to the framerate of the video. Depending on the subsequent analysis the overlap should be varied. As an example (Heinzel et al. 2002) proposes that overlap should be chosen to preserve flatness and minimise computational effort. They note that a pragmatic solution is to use a 50% overlap which works well in the situations where the spectral impulse response of the system is unknown. Consequently, for this work we chose a 50% overlap.

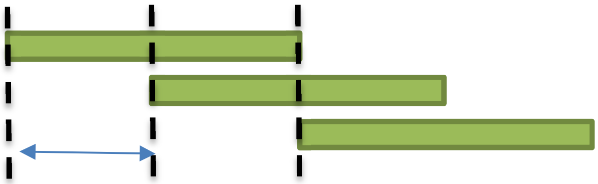


Figure 10: Windowing the audio signal into frames

After the initial windowing the individual frames are combined to know when human speech is occurring or not. (Morrison et al. 2007) proposed the use of endpoint detection using the energy contour of the frames along with the zero crossing rate. Using this approach individual frames can be marked as belonging to speech and passed for subsequent processing or not and discarded. The specific way in which the frames are processed is shown in Figure 11. After the signal is segmented into segments with speech within them is classified as described in D3.1.

[5] https://www5.cs.fau.de/de/mitarbeiter/steidl-stefan/fau-aibo-emotion-corpus/

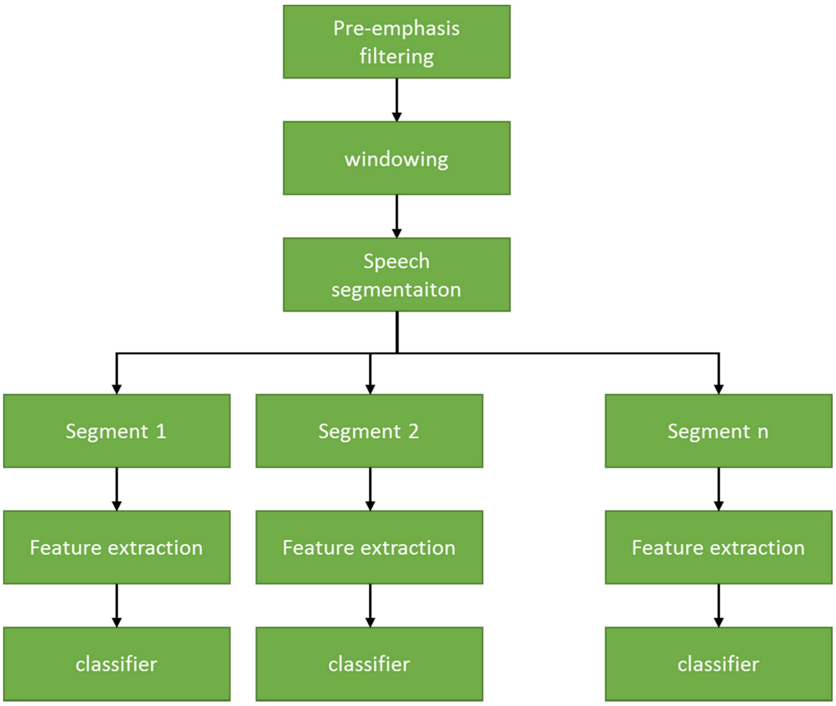


Figure 11: Processing continuous speech to produce classified emotional utterances

In April 4 to April 6 2016 the existing voice capture and analysis infrastructure was part of a small scale trial in two schools in Verona, Italy. After consultation with the relevant parties it was decided that the students playing the game would be seated around a table playing the game on tablets as shown in Figure 12. This layout is considered to promote cooperation and improve the learning outcomes.

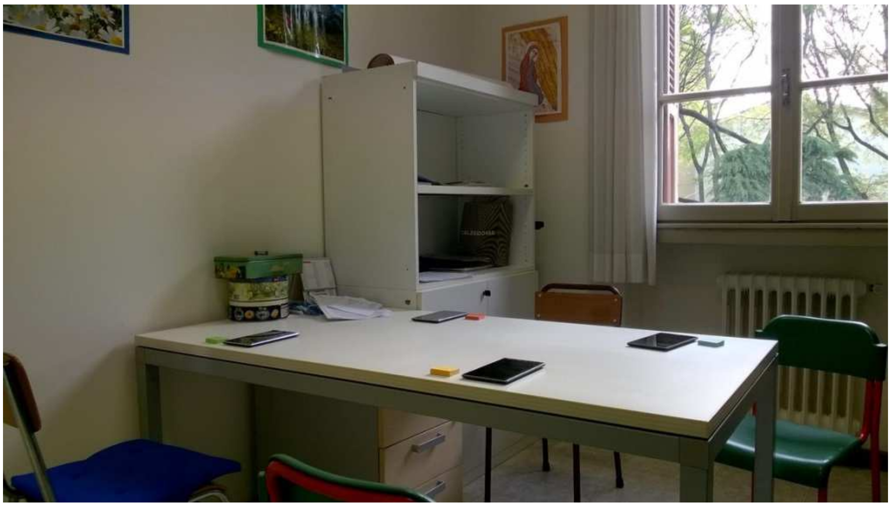


Figure 12: The arrangement of students when playing the game.

This arrangement however leads to a problem where we can perceive crosstalk between players. This is because all microphones can perceive the audio from all the players. In order to reduce this problem signal source separation needs to be applied to the audio streams. An example of the captured audio in all four tablet is illustrated in Figure 13. Note that there is significant correlation between the signals.

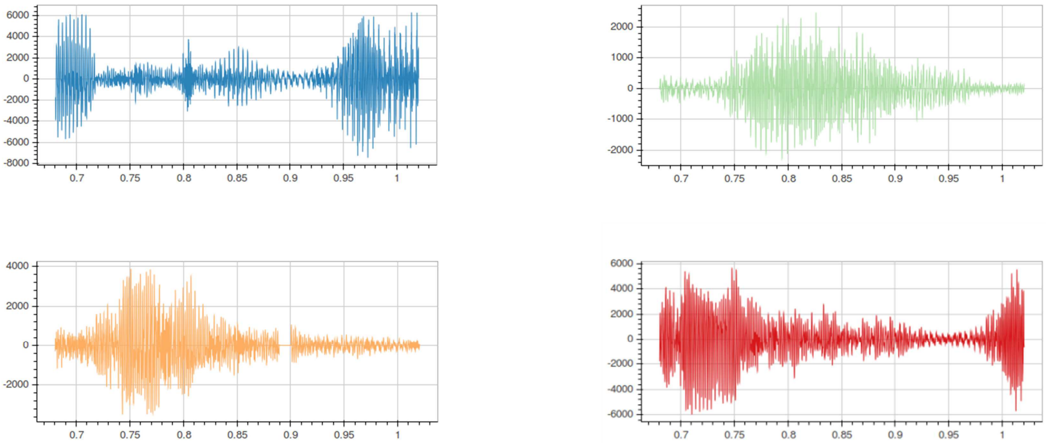


Figure 13: The audio at the same time across 4 different devices

The most popular approach in which the individual signals can be decoupled is independent component analysis (ICA) (Choi et al. 2005). There are a number of assumptions that this technique assumes. Firstly, the original signals are independent of one another. In the case of microphone recordings this is the case so long as the number of microphones is equal to the number of speakers. Secondly, the underlying distributions are non-Gaussian which appears to be the case for general speech signals. (Gazor 2003) suggests that the underlying distribution is Laplacian. Finally, the individual samples need to be synchronized. This later point is discussed in section 5.1 and almost certainly makes it difficult to employ ICA. An alternate algorithm that aligns the signals temporarily is called DUET (Rickard 2007). Due to the weak synchronization this is considered a better algorithm to disambiguate the voice signals.

4.2 Visual information coming from facial analysis

4.2.1 Facial expressions

Facial motion plays a major role in expressing emotions and conveying messages. The analysis of facial expressions for emotion recognition requires the extraction of appropriate facial features and consequent recognition of the user's emotional state that can be robust to facial expression variations among different users. Features extracted by applying facial expression analysis techniques can range from simply geo-locating and calculating actual anthropometric measurements, to summarizing an entire group of feature-group elements under a single emotional category, such as happiness or surprise. Using sophisticated and well-trained shape and landmark tracking techniques, specific facial feature points can be identified and located for every consecutive frame obtained by a camera-like sensor. Early forms of low level data processing can then be applied to identify and track muscle activity into specific Action Units (AUs). These AUs can be seen as a form of mid-level representation of the raw data.

In this respect, we followed the approach described in (Soleymani et al, 2012) in which landmark processing leads to low-level facial features describing the three most expressive regions of the human face: the upper component, the middle component and the lower component. Subsequently, we distincted our extracted AU features in two categories, mainly upper face and lower face AUs. In order to extract the aforementioned set of AUs, we followed a similar approach as (Tian et al, 2001), which incorporates the feature tracking capabilities offered by a dense-ASM tracking framework. More specifically, we employed two three-layer neural networks with one hidden layer to recognize AUs through a number of parameters defined by low-level features extracted for the upper and lower face regions. The ultimate goal of identifying and extracting AUs is to classify expressions under a certain emotion category.

For comparison reasons, we formalized our results against the baseline method of the FERA 2011 challenge in order to see how our neural networks performs on an entirely different dataset. The results of the AU detection, measured using F1-score for direct comparison of our approach against the FERA 2011 baseline method (Valstar et al, 2011) and the corresponding reported results of a naïve AU detector, are depicted in Figure 14.

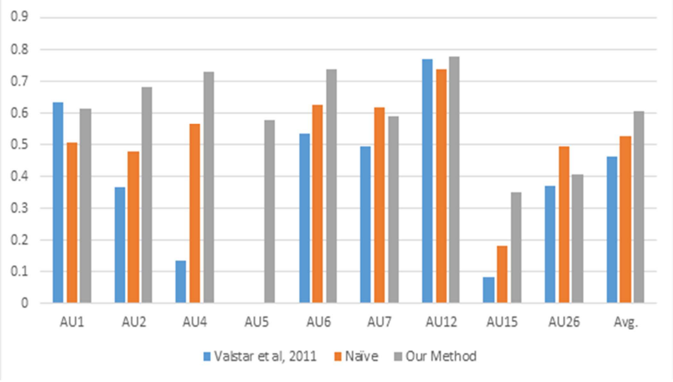


Figure 14: Comparison of the proposed AU detection method (F1-score) against the FERA 2011 baseline method and a naive AU detector

M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic, "A multimodal database for affect recognition and implicit tagging", Affective Computing, IEEE Transactions on, *3*(1), pp. 42-55. vol. 2., pp.68-73, 2012

Y. L. Tian, T. Kanade, J.F. Cohn, "Recognizing action units for facial expression analysis", Pattern Analysis and Machine Intelligence, IEEE Transactions on, 23(2), 97-115, 2001.

M. F. Valstar, B. Jiang, M. Mehu, M. Pantic, and K. Scherer, "The first facial expression recognition and analysis challenge", In Automatic Face & Gesture Recognition and Workshops, IEEE International Conference on, pp. 921-926, 2011.

4.2.2 Gaze analysis

As described in Deliverable D3.1, multiple levels of feature descriptors are extracted, with regards to raw gaze pattern measurements as well as indications on higher level cognitive processes, such as engagement and attention. In contrast to the Facial expression analysis features described in the previous sub-Section, the fusion algorithms will gain access only to raw gaze pattern measurements, as according to literature higher concepts do not show direct link to specific emotions.

In a similar way to facial expression feature analysis, determining visual characteristics such as head pose and the direction of a user's gaze are a vital part of this kind of feedback. In this respect, we can use the position and movement of prominent points around the eyes and the position of the irises to reconstruct vectors which illustrate the direction of gaze and head pose. These vectors will be used as an indication of whether the user is currently *attentive*, i.e. looking into the screen or not and, in conjunction with our gaze tracking system, whether the users' eyes are fixed at a particular spot for long periods of time. Prosocial affect fusion algorithms will blend gaze information with face data to get an indication of whether the game attracts their attention.

4.3 Visual information coming from body motion analysis

4.3.1 Body motion

Extracting body motion analysis features that can be fused along with the data acquired from visual and audio cues is a challenging task. Furthermore, body motion analysis data are crucial in generating multi-modal data in gameplay environments where players' facial analysis data are noisy or even missing.

In a similar way to facial expression features, the fusion algorithms will process either low-level feature group or high-level features in order to reach a decision on the player's prosocial affective state. The first group includes features such as kinetic energy, fluidity, symmetry, which could be extracted and used in real time by the fusion process. On the other hand, the latter presupposes the creation of a time window, where the classifier can process and analyze the motion characteristics, making a decision about the action (emotion) performed by the user. Most of the body features described in Deliverable D3.1 are extracted through joint-oriented skeleton tracking using depth and RGB information from Kinect sensor. More specifically, the extracted features are classified into the following broad categories: i) kinematic related features: kinetic energy, velocity and acceleration, ii) spatial extent related features: bounding box, density and index of contraction, iii) smoothness related features: curvature and smoothness index, iv) symmetry related features: wrists, elbows, knees and feet symmetry, v) leaning related features: forward and backward leaning of a torso and head as well as right and left leaning and vi) distance related features: distances between hands, distance between hand and head as well as hand and torso. An example of the kinetic energy measurement during the play of the "Path of Trust" prosocial game (Apostolakis et al, 2015), is demonstrated in Figure 15.

For the combination of different set of features, we designed a two-layered network in which we have stacked seven NNs, six at the first layer and one at the second layer. Each layer is trained separately, starting from base layer and moving up to the second, with no feedback from the higher layer to the lower layer. Each NN of the first layer receives as input the features of a different group of features. Then, the output probabilities of the first layer are fed as input to the second one and a separate NN is trained (Kaza et al, 2016).

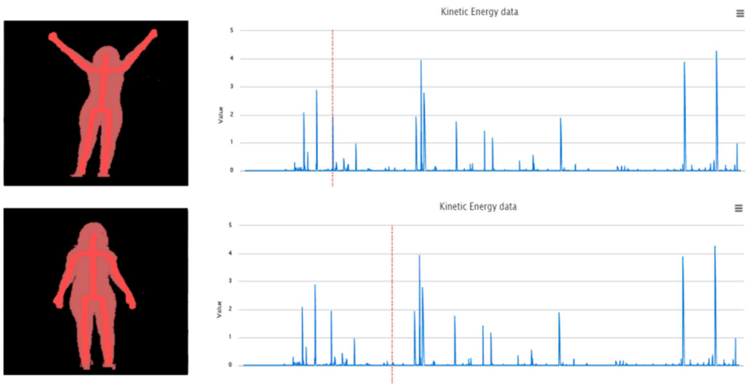


Figure 15: Kinetic energy data measurement using the Kinect sensor during "Path of Trust" gameplay. The continuous blue line indicates Kinetic energy measurements over time through the entire session. The dotted red vertical line indicates the current frame. Top image depicts density calculation when user's body spatial extent is increased through the extension of the hands. Bottom image shows the corresponding measurement when the student's body is contracted.

In order to evaluate the performance of body mono-modal classifier, we created a dataset containing Kinect recordings of body movements, which express the 5 basic emotions that are likely to appear in a gameplay scenario. The proposed deep learning network classifier outperformed a number of state of the art classifiers with a recognition rate of 93%. The detailed results of these comparative study are presented in Figure 16

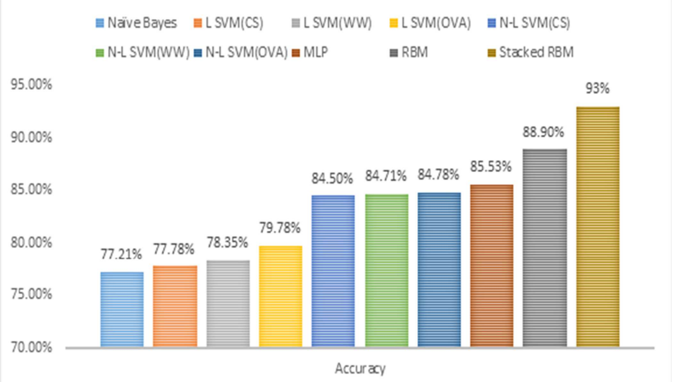


Figure 16: Comparison of the proposed body motion analysis algorithm for emotion recognition against a number of state of the art classifiers.

K. Apostolakis, K. Kaza, A. Psaltis, K. Stefanidis, S. Thermos, K. Dimitropoulos, E. Dimaraki, P. Daras, "Path of Trust: A prosocial co-op game for building up trustworthiness and teamwork", InB Games and Learning Alliance: Fourth International Conference, GALA 2015, Rome, Italy, December 9-11, 2015.

K. Kaza , A. Psaltis , K. Stefanidis , K. Apostolakis , S. Thermos , K. Dimitropoulos, P. Daras, "Body Motion Analysis for Emotion Recognition in Serious Games", HCI International 2016, Toronto, Canada, 17 - 22 July 2016.

4.3.2 Hand motion

Analysis of arm movements has shown that, considering a dimensional emotional space represented by measures for valence and arousal, the velocity, acceleration, and jerk of the hand movement is highly correlated with the arousal component. Thus, features related to the user's motion of the hands cannot be used for emotion fusion process.

4.4 Comparison of emotions

In order to use the outputs from the emotional classifiers in the fusion stages, the emotions need to be brought into a consistent representation. This representation and the underlying model need to obey a number of mathematical properties. Namely, one in which each emotion has a unique mapping into the space and that emotions in the space correspond to changes in emotional state. Furthermore, the space needs to be metric so that distances in the space are defined for any pair of emotions, are symmetric, and obey the triangle inequality. The space should allow traces of emotions over time. This leads to an ideal that small changes in the values of the space should lead to related emotions. Before making concrete proposals this section will briefly review the existing emotion models.

4.4.1 A brief overview of Models of emotion

Classically there are two views of emotions. These are discrete or categorical models and dimensional models. In discrete emotional models, all people are considered to have a set of innate emotions. It further views these emotions to be fundamental (like atomic particles) and consequently exist across cultures. The most famous of these models is due to (Ekman 1992). Ekman proposed a model consisting of six basic emotions: anger, disgust, fear, happiness, sadness, and surprise. Generally, these models are supported by the study of language (Bann & Bryson 2013) with the view that emotions are influenced by our ability to express them.

An example of the Ekman model is shown in Figure 17. There are two specific emotions illustrated. As the emotions are categorical in nature you can experience a mix of these simultaneously. Under this sort of model the emotional space could be considered a vector with real valued elements. These models are simple to understand but suffer from lack of extensibility. Addition of an emotion is simply via the addition of an element to the vector. However, this treats all emotions are unique where in reality a number of emotions are related. For example, “angry” and “annoyed” are very related. This inability to disambiguate emotions makes it a non-ideal representation. Consequently, discrete models are not a good fit for a unifying representation for emotions.

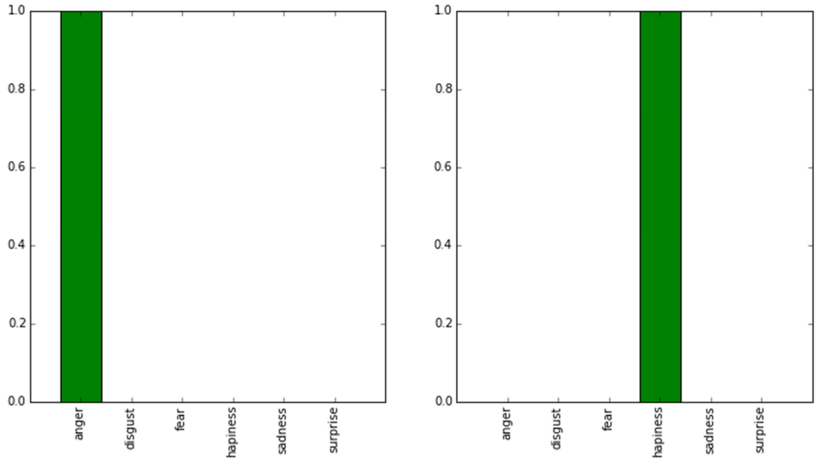


Figure 17: Ekman model of emotion (left) angry (right) happy

Dimensional models express emotions as being made up of values aligned with more or more axes. These are the historical view with an underlying notion that that a complex neurophysiological system in the brain gives rise to all possible emotions. Typically the axes include valance, arousal, and a number of other parameters such as pleasure, arousal, or pleasantness. These models most commonly have two dimensional axes but unidimensional models have been proposed.

The circumplex model of (Russell 1980) implies a circular interpretation to emotional states. Grounded in a neurophysiological model where separate valance and arousal circuits in the brain combine to produce emotional responses. Values in the circumplex are usually considered in terms of their angle about the origin and the magnitude of the emotion. As a consequence this model is usually considered as a circle about the origin. An image showing a number of emotions and their position in space is shown in Figure 18.

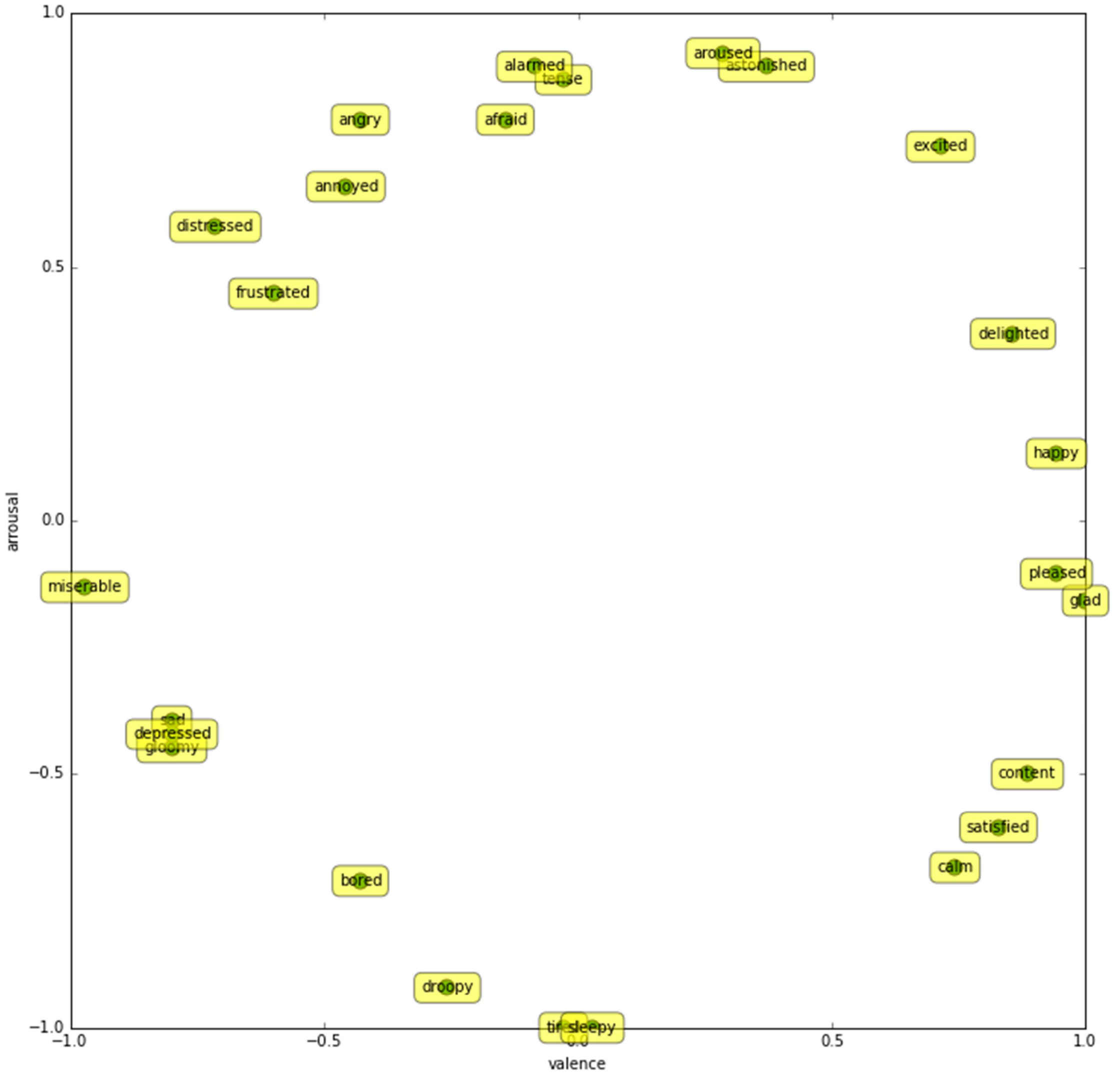


Figure 18: Circumplex model of emotion

Another circular model of emotion is attributable to (Plutchik 2001). It illustrated in Figure 19. It consists of four basic emotions and their opposites. Increased intensity emotions were along the same axis as the basic emotions. This allows for multiple rings of emotion. Furthermore, by drawing an analogy with colour wheels he allowed emotions to be mixed. Thus, for example, anticipation plus joy was equal to optimism. Unfortunately, the model does not support addition of different intensity of emotions. So, for example, interest plus joy is also equal to optimism. This additive approach makes it difficult to generally define arithmetic on this model.

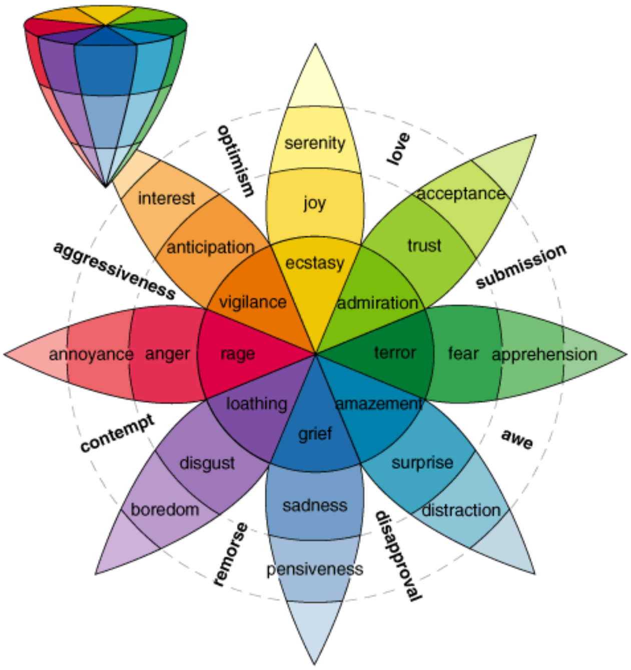


Figure 19: Plutchik wheel of emotion

A more recent work on bringing emotions together was presented in (Cowie et al. 1999). Subjects were asked to rate different emotional words using a similar emotional space as proposed by (Russell 1980). Analysis of a number of subjects leads to a basic English vocabulary for emotion and a series of schema describing them.

Vector based models of emotion define a specific position in space associated with each emotion. The first such model was (Bradley et al. 1992). The defining dimensions of this model were arousal and pleasantness. This is a rotation of the space employed in the circumplex model.

It may be that two dimensional models are insufficient to describe emotional spaces. An example of this is given by the Pleasure-Arousal-Dominance (PAD) model of (Mehrabian 1996). This is illustrated in Figure 20. Multi-dimensional spaces may represent the emotional space more accurately but getting data for them is difficult. Generally the approach involves use surveys of the target group under question. This is difficult for us to perform in practice as would require domain experts to perform the analysis. Furthermore, it would need different experts for different languages.

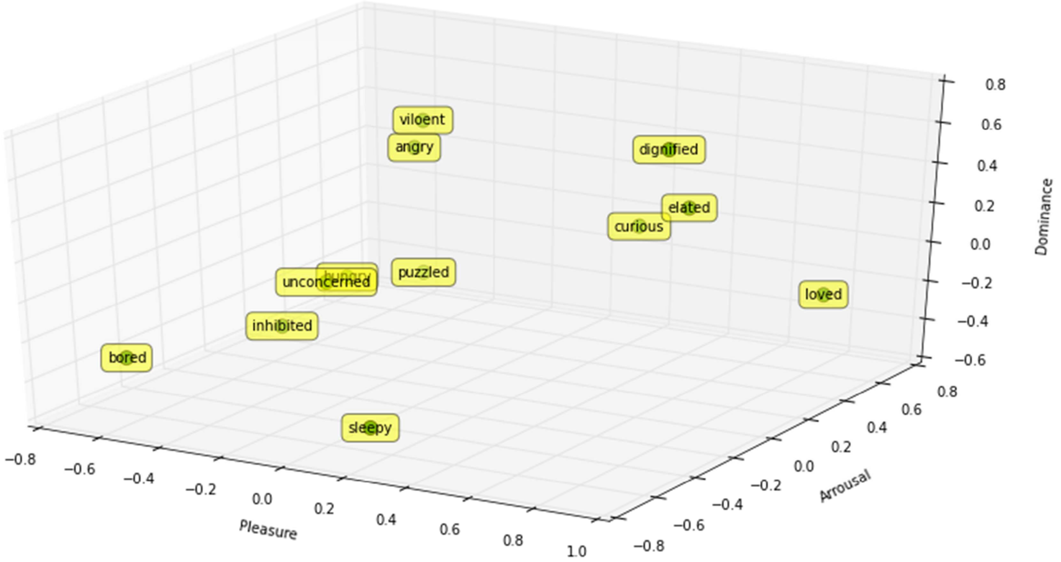


Figure 20: Pleasure-Arousal-Dominance (PAD) emotional space

There are a number of different models presented in this section to represent emotions. The two dimensional models are more useful to our application as they have an intuitive visual interpretation. Additionally, similarities between emotions can be performed via distance measures.

4.4.2 Unifying the emotional descriptions

It is proposed that the underlying representation for emotions is within a dimensional space. This, as mentioned, will allow a number of useful mathematical properties which can be exploited within the fusio and other parts of the project. Specifically we chose the Valance-Arousal space. The reason was the ANEW project (Bradley & Lang 1999) which documents the valance and arousal of approximately 1000 english words. Examples for the Ekman categories are shown below in Table 8. In terms of the Valance-Arousal space the emotional categories are defined by a Gaussian distribution with mean (μ) and standard deviation (σ).

Category	Valence (μ)	Valence (σ)	Arousal (μ)	Arousal (σ)
Anger	2.34	1.32	7.63	1.91
Disgust	2.45	1.41	5.42	2.59
Fear	2.76	2.12	6.96	2.17
happiness	8.21	1.82	6.49	2.77
Sadness	1.61	0.95	4.13	2.38
Surprise	7.47	1.56	7.47	2.09

Table 8: Mapping from category to valence and arousal values. Means (μ) and standard deviations (σ) are given.

Using this assumption we can plot a figure which illustrates the position in the Valance-Arousal space for each of the categories. This is depicted in Figure 21. Note that while the original data had values in the range [1, 9], the figure has mapped them to [-1, 1] via a linear transform. This remapping makes neutral emotions cluster around the origin which is advantageous for reasoning over the space.

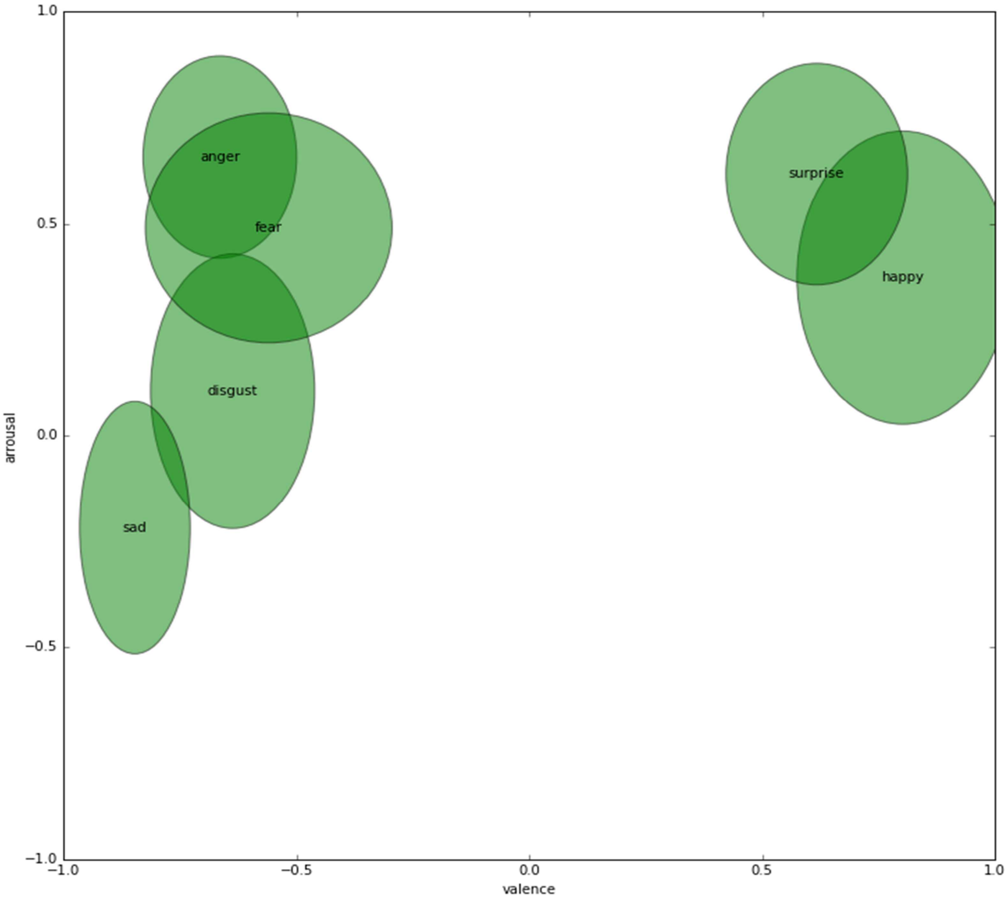


Figure 21: Ekman catgories mapped in Valance-Arousal space.

Mapping from categories to the Valance-Arousal space will be performed by drawing values from the underlying distribution that corresponds to the category. The reverse mapping can be performed using Mahalanobis distances. Thus we will use Valance-Arousal values as an intermediate representation for emotions. This representation frees us from the use of categories and has the nice mathematical properties. Concrete examples of this are illustrated in section 5.2.

4.4.3 Recommendations

It is considered that the best way to unify the emotions in the project is to use a dimensional space to represent the emotions. This has multiple advantages over using the categorical models. The most important of which is the ability to measure changes in this space as contrasted with categorical models. Due to the prevalence in the literature, valance-arousal spaces will be used for this.

4.5 Comparison of Engagement

An important step towards creating an adaptation mechanism is to understand the relationship between game mechanics and features contributing to the learning effectiveness of a game. Such understanding would allow measuring different features at fusion stage of the prosocial platform that in turn can feed back the game adaptation process. (Olsen et al. 2011) proposes an approach for measuring effectiveness for learning where effectiveness is seen as a collective measure of usability, playability and learning outcome. In order to have any reliable measure of playability, some basic level of usability need to be there. Furthermore, no learning outcomes can be achieved unless there is some level of playability present. Resnick et al. define playability as “the entertainment without fear of present or future consequences; it is fun” (Resnick & Sherer 1994). There aren’t well developed and used measures for playability; it is measured by using the developed scales for immersion, presence, flow and engagement (Olsen et al. 2011). These can be conceptualized as representing a progression of ever-deeper engagement in game-playing.

Engagement is an essential element of the player experience. According to (Lehmann 2012) user engagement is the quality of the user experience that emphasizes the positive aspects of the interaction, and in particular the phenomena associated with being captivated by a game, and so being motivated to use it. Successful games are not just played, they are engaged with; players invest time, attention, and emotion into them. In an environment where pupils display quite often splitting attention problems, it is essential that game industry design engaging experiences. So-called engagement metrics are commonly used to measure game player engagement. Various methods have been described in literature to measure engagement.

Immersion is typically used to describe the experience of becoming engaged in the game-playing experience while retaining some awareness of one’s surroundings (Banos 2004; Singer & Witmer 1999)**.** It is likely that most regular game players experience some degree of immersion.

Presence has been commonly defined in the terms of being in a normal state of consciousness and having the experience of being inside a virtual environment (Tamborini & Skalski 2006). Most, but not all video game players are likely to have the capacity to experience presence, given the appropriate conditions.

Flow is the term used to describe the feelings of enjoyment that occur when a balance between skill and challenge is achieved in the process of performing an intrinsically rewarding activity (Moneta & Mihaly Csikszentmihalyi 1999). Flow states also include a feeling of being in control, being one with the activity, and experiencing time distortions. Because it involves experiencing an altered state, the flow experience may be somewhat less common than immersion or presence.

Psychological **absorption** is the term used to describe total engagement in the present experience (Irwin 1999). In contrast to immersion and presence, and in common with flow, being in a state of psychological absorption induces an altered state of consciousness. Becoming involved while forget about themselves and their environment and experience the narrative as if it was real and being part of it.

4.5.1 Some characteristics associated with user engagement

Player engagement possesses different characteristics depending on the game; e.g. how users engage with a single player or a multiplayer game is very different. However, the same engagement metrics are typically used for all types of player, ignoring the diversity of experiences. In addition, discussion on the “right” engagement metrics is still going on, without any consensus on which metrics to be used to measure which types of engagement. In the following we will try to demonstrate the diversity of user engagement, through the identification and the study of models of player engagement.

In a recent study, (Attfield 2011), suggested the following characteristics associated with user engagement.

Being engaged in an experience involves **focusing attention** to the exclusion of other things, including other people. There is a relation between subjective perception of time during gameplay and the level of player engagement. The more engaged someone is, the more likely they are to underestimate the passage of time. Focusing attention could possibly be measured by questionnaires, follow-on tasks and gaze tracking algorithms.

O'Brien defines engagement as "a category characterized by positive affect, where engaged users are affectively involved" (O'Brien & Toms 2008). **Affect** relates to the emotions experienced during interaction, and could be measured in real time using physiological sensors such as facial emotion detection and body emotion detection.

Aesthetics concerns the sensory, visual appeal of an interface and is seen as an important factor for engagement. Some players became engaged by the layout or aesthetics of the game. They talked about being attracted to graphics, music and features that first caught their attention. Furthermore, interactive experiences can be engaging because they present users with novel, surprising, unfamiliar or unexpected experiences. Novelty appeals to our sense of curiosity, encourages inquisitive behaviour and promotes repeated engagement. Such reactions could be captured by facial and body expression recognition in combination with gaze tracking algorithms.

Richness captures the growth potential of an activity by assessing the variety and complexity of thoughts, actions and perceptions as evoked during the activity (e.g., variety, possibilities, enjoyment, excitement, challenge). Body sensors (Kinect), hand motion sensors (Leap Motion) and other input tracking devices such as mouse and keyboard, could be a reliable indicator of the level of richness experienced.

In the table below, we summarise the identified characteristics of user engagement presented in the previous paragraph, highlight their possible ways to objectively measure them.

Characteristic	Definition	Measures
Focusing Attention	Focusing attention to the exclusion of other things	Gaze tracking, follow-on tasks, Questionnaires
Positive Affect	Emotions experienced during interaction	Face & Body emotion Detection
Aesthetics	Sensory and visual appeal of an interface	Face & Body expression Recognition, Gaze tracking
Novelty	Novel, surprising, unfamiliar or unexpected experiences	Face & Body expression Recognition, Gaze tracking
Richness	Levels of richness	In game activity, mouse clicks

Table 9: Characteristics of user engagement and possible measures

4.5.2 A brief overview of Models of engagement

Having defined user engagement and elaborated some of its main characteristics we now look into potential approaches to its assessment.

User experience evaluation metrics can be divided in two main groups: subjective and objective. Subjective measures record a user's perception, generally self-reported questionnaires. User's subjective experiences are central to user engagement and we consider methods for assessing these. Subjective experiences, however, can have objectively observable consequences, and so we consider objective measurements may be indicative of user engagement. These include independent measures such as the passage of time or number of mouse clicks to complete a task.

In the first group, post-experience questionnaires, interviews and tests are used to elicit user engagement attributes or to create user reports and to measure engagement in relation to a given game experience. They can be carried out within a lab setting, or via on-line mechanisms (including crowd-sourcing). Such an instrument was developed by (Brockmyer 2009)**.** They developed a game engagement questionnaire which measures the levels of engagement when playing games. Engagement is seen as passing through several stages from low to high engagement (Brockmyer 2009). These stages are immersion, presence, flow and absorption where immersion indicates the lowest levels of engagement and absorption is associated with the highest levels of engagement. The questionnaire has the potential to identify the different levels of engagement when playing a game. In a more recent study (Whitehill 2014), human annotators were instructed to label clips/images for "How engaged does the subject appear to be". They have followed an approximate scale to rate engagement, where "Not engaged at all" indicated the lowest levels, while "Very engaged" is linked to highest levels of engagement. (Attfield 2011) noted that it is not straightforward how to produce a general purpose user engagement questionnaire, some characteristics may generalize well, and others may not. Thus, we need to generate new user engagement instruments relevant to specific kinds of interaction and user. Subjective methods have known drawbacks, are not sensitive to ways in which an interaction changes over time. In this case questionnaires may not be the best tool, and objective measures seem better suited.

The second group uses task-based methods (follow-on task), and physiological measures to evaluate the cognitive engagement (e.g. facial expressions, vocal tone, body activity) using tools such as gaze tracking, face and mouse tracking. In game data such as task duration, task accomplishment and other task related events could be indicative of player engagement. The performance on a side quest task immediately following a period of engaged interaction is something that could be used a measure of cognitive engagement. Game researchers have found that the more engaged the person is during gameplay, the longer it takes them to complete the unrelated side quest afterwards (O'Brien & Toms 2008). In contrast, physiological data could be captured by a broad-range of sensors (Kinect, camera, microphone, Leap motion) are related to different affective states. For example, a camera could capture gaze changes (related to attention, strong emotion, difficulty) and facial muscle changes (related to positive or negative affect). In addition, mouse and keyboard inputs could capture stress and certainty of response, while Kinect and leap could capture actions related to boredom and fun. Such sensors have several advances over questionnaires, since they are more objective and they are continuously measured while there is a direct connection with the emotional state of the user. In general, such measures could be highly indicative of engaging states through their links with attention, affect, perception of aesthetics and novelty.

4.5.3 Recommendations

Considering what has been described above, the best way to acquire a quantitative indicator related to engagement is to use task based metrics implemented in game scenarios, which measure the level of accomplishment, or the duration of specific game quest, fused with vision-based facial and motion analysis data captured by sensors in a control environment.

5 Multimodal Fusion

This section approach used within ProsocialLearn to perform fusion of the observations described in the previous section. It also discusses the problem of synchronization within the platform.

5.1 Synchronisation

Despite their beneficial effect, multimodal fusion methods come with a certain cost and complexity in the analysis process. This is due to the diversity of characteristics of the modalities involved. The fact that different media are usually captured in different formats and rates, make it hard to represent the time synchronization between the multimodal features. For example, a web camera captures image sequences at a frame rate which may be quite different from the rate that a microphone captures sound samples. Therefore, a pre-processing part of the fusion module should deal with the asynchronous observations to better accomplish the task. Another thing to be considered when selecting the fusion strategy is the dissimilarity of the processing time of different types of media streams.

The time when the fusion must be carried out is an important consideration in this multimodal task. Certain characteristics of sensors, such as varying data capture rates and processing time of the sensor, poses challenges on how to synchronize the overall process of fusion. Often this has been addressed by performing the multimedia analysis tasks (such event detection) over a timeline (Chieu & Lee 2009). A timeline refers to an actual picture of events happened in a certain period of time, containing important information for the examined task. The timeline-based accomplishment of a task requires identification of events at which fusion of multimodal features should take place. Due to asynchrony and diversity among streams and because of the fact that different analysis task are performed at different granularity levels in time, the identification of these events, i.e. when the fusion should take place, is a challenging issue (Atrey et al. 2006).

As the fusion can be performed at the feature as well as the decision level, the issue of synchronization is also considered at these two levels. In the feature level synchronization, the fusion scheme integrates the unimodal features captured at the same time period, before learning concepts (Chetty & Wagner 2005). On the contrary, the decision level synchronization needs to determine those events along the timeline at which the learned decisions from all unimodal features are integrated to learn higher concepts. However, in both levels of fusion, the problem of synchronization arises in different forms.

Figure 22a illustrates the synchronization at the feature level between two different types of modalities. At the time t =1, the fusion process receives raw data from both modalities. Next, these arbitrary data are processed in order to derive numerical features. The processing time for the feature extraction differs for each modality (e.g. 2 and 1.5 time units from modality 1 and modality 2, respectively in Figure 22). Due to the different time periods of the data processing and feature extraction, when these two features should be combined, remains an issue. One could follow a simple strategy to solve that issue, by fusing the derived features at regular intervals (Atrey et al. 2006). This strategy appears to be computationally less expensive and an alternative strategy could be followed which combines all the features at the time instant they are available (e.g. at t =3 in Figure 22).

The decision level synchronization has been illustrated in Figure 22b. Unlike the previous fusion scheme, where the errors in synchronization were due to feature extraction processing time, the error grows as added extra time in the decision making process. For example, as shown in Figure 22b, the time taken in obtaining the decision could be 1.5 and 1.75 time units for modality 1 and modality 2, respectively. However, fusing the obtained decisions could be done using different strategies as for example the time instant all the decisions are available, (t =4 in Figure 22b). Different strategies should be adopted to match each multimodal fusion process.

Another important synchronization issue is to determine the amount of raw data needed from different modalities for accomplishing a task. To mark the start and end of a task (e.g. event detection over a timeline), there is a need to obtain and process the data streams at certain time intervals. For example, from a video stream of 25 fps, less than a second of data (10 frames) could be sufficient to determine a human facial emotional expression event (by computing the facial muscle displacement in a sequence of images); however the same event (body emotional gesture) could be detected using 3 seconds of Kinect[6] stream data of 30 fps. This time period, which is basically the minimum amount of time to accomplish a task, could be different for different tasks when accomplished using various modalities. Ideally, it should be as small as possible since a smaller value allows task accomplishment at a finer granularity in time. In other words, the minimum time period for a specific task should be just large enough to capture the data to accomplish it.

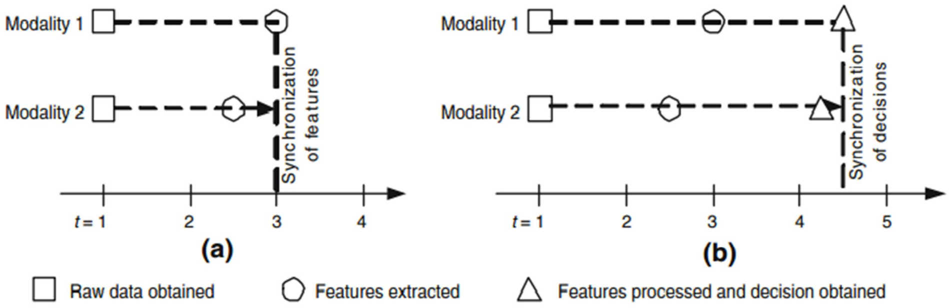


Figure 22: Illustration of the synchronization between two modalities at (a) feature level (b) decision level

Clock synchronization[7] is a problem from computer science and engineering which deals with the idea that internal clocks of several computers may differ. Even when initially set accurately, real clocks will differ after some amount of time due to clock drift, caused be clocks counting time at slightly different rates. Network Time Protocol[8] (NTP) is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks. NTP is intended to synchronize all participating computers within a few milliseconds of Coordinated Universal Time (UTC). It uses an algorithm to select accurate time servers and is designed to mitigate the effects of variable network latency. NTP can usually maintain time within tens of milliseconds over the internet, and can achieve better than one millisecond accuracy in local area networks under ideal conditions. Asymmetric routes and network congestion can cause errors of 100ms or more. NTP uses tree-like topology, but allows you to connect a pool of peers for better synchronization on the same strand level. This is ideal for synchronizing clocks relative to each other. For better relative and absolute clock synchronization, one has to run its own NTP server.

[6] https://en.wikipedia.org/wiki/Kinect

[7] https://en.wikipedia.org/wiki/Clock_synchronization

[8] https://en.wikipedia.org/wiki/Network_Time_Protocol

The accuracy of NTP is of the order of tens to hundreds of milliseconds. There are other systems which provide more accurate time such as Precision Time Protocol (PTP)[9] or Synchronous Ethernet. Specifically, PTP has been shown to achieve microsecond accuracy on the synchronization of the clocks. However, the use of these protocols require hardware to be aware of the protocols. For instance, PTP requires the network equipment to have clocks which are used to update the clock data in the network packet. This is certainly not guaranteed when you are using distributed clients across the internet.

All these approaches however suffer from the problem that the time is not monotonically increasing. This means that it is possible for a sequence of events to reverse their order relative to one another due to clocks correcting for drift.

In ProsocialLearn we adopted a pragmatic approach to the synchronization of the clocks. This started with an acceptance that we cannot get perfect synchronization between the various components of the system (clients and platform). With this we examined the accuracy of various components. An example is shown in Table 10: Typical inter-sample rate of different modalities. Note the values are rounded to the nearest close values.

Device	Time between samples (s)
Video	0.01
Audio	0.0001
Game information	1

Table 10: Typical inter-sample rate of different modalities.

With this simple back of the envelope exercise we cannot expect the ProsocialLearn platform to achieve an accuracy of more than the worst component which is the game traffic. This also means that we can batch samples up on the client side before sending them to the server. This, allows us to use NTP as a time synchronization protocol as it achieves the required accuracy.

So the final adopted approach is as follows:

- Use NTP to synchronize the clocks on all devices (bother the platform and the clients)
- Record NTP timestamps on the client for samples
- Record NTP timestamps on the server when the data is received

By combining the timestamps and computing offsets we can achieve synchronization with the required accuracy. Additionally, we can use the two independent measures of time to correct for drift correction.

5.2 Multimodal fusion in ProsocialLearn

A common distinction when defining multi-modal approach considers if the fusion happens at feature or at decision level. We have already presented these possibilities when introducing the problem of synchronization in Section 5.1.

[9] https://en.wikipedia.org/wiki/Precision_Time_Protocol

Feature level fusion works well when the information produced by different sensors is consistent and compatible. When signals different in nature are present is instead better to adopt a decision level approach that is computationally lighter and predicates on consistent space (in our case the emotion space). The different nature of video and audio signals suggests therefore the use of a decision level multimodal fusion for the emotion classification.

As shown in Figure 23 a number of different sensors will first, undergo feature extraction and then passed through classifiers. When the modalities for the sensors are complementary these will combined in the classifier. This is a concrete example of feature fusion. The individual classifiers can output either emotional labels or a value in the valence arousal space (as described in Unifying the emotional descriptions 4.4.2). In all cases the outputs need to use a common representation. For this purpose a mapper from labels to valance-arousal space will be used. Individual measurements will then be combined via decision level fusion to decide the outcome. The remainder of this section will give a concrete example of decision level fusion for body motion and face. It is considered that these modalities are complementary as emotional expressions are played out over the whole body and not just the face.

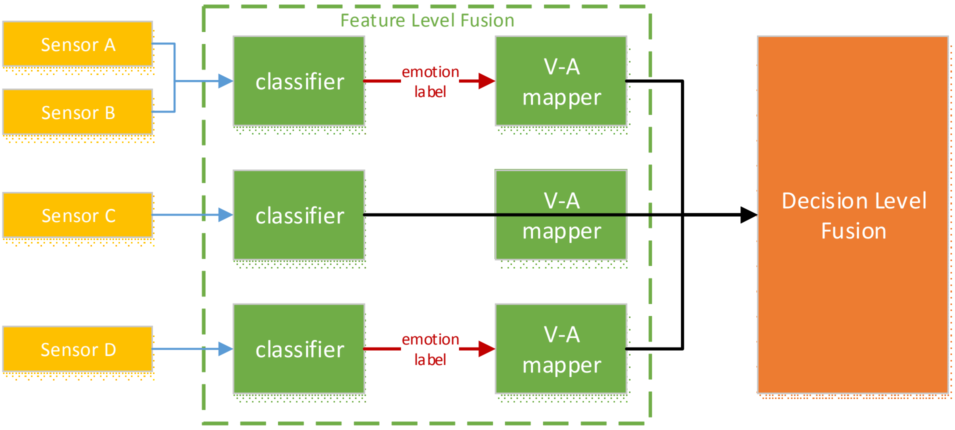


Figure 23: Breakdown of how decision level fusion will interact with emotion detection

5.2.1 Multi-Modal Fusion based on Facial Expression and Body Motion Analysis

We developed a multimodal fusion architecture that uses stacked generalization on augmented noisy datasets and provides enhanced accuracy as well as robustness in the absence of one of the input modalities. Moreover, we designed a list of body actions and facial expressions commonly encountered in a typical game, eliciting emotions based on Ekman’s discrete categorization theory. Based on this list of emotions, a bimodal database was created using Microsoft’s Kinect sensor, containing feature vectors extracted from users' facial expressions and body gestures.

Facial Expression Stream

As described in D3.1, we separated our extracted AU features into two categories, mainly upper face and lower face. More specifically, we employ two three-layer neural networks with one hidden layer to recognize AUs through a number of parameters defined by low-level features extracted for the upper and lower face regions. The ultimate goal of identifying and extracting AUs is to classify expressions under a certain emotional category. We further concatenate the two neural network posteriors in a unified representation, and train an extra layer on top of them as shown in the left part of Fig.2.

Body Motion Stream

In order to combine information extracted from body stream, we propose a two-layered network in which we have stacked seven NNs, six at the first layer and one at the second layer. Each layer is trained separately, starting from base layer and moving up to the second, with no feedback from the higher layer to the lower layer. Each NN of the first layer receives as input the features of a different group of features. Then, the output probabilities of the first layer are fed as input to the second one and a separate NN is trained. The output probabilities of the second layer constitute the classification result of the body motion analysis mono-modal classifier as shown in the right part of Figure 24.

Dynamic Fusion

In the version 1 of D3.2 (see Appendix C), we have presented that deep learning networks can be applied at feature level as well as at decision level, being trained directly on raw data or decisions accordingly. In this direction, we employ a late fusion scheme, where each intermediate classifier is trained to provide a local decision. In terms of affect, local classifiers return a confidence as a probability in the range of [0, 1] in a set of predefined classes. The local decisions are then combined into a single semantic representation, which is further analyzed to provide the final decision about the task. The aforementioned scheme for late fusion is illustrated in Figure 24.

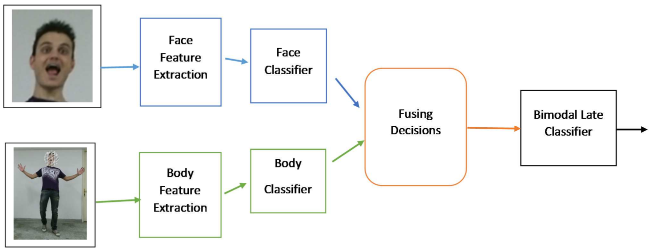


Figure 24: Bi-modal late fusion scheme

In the proposed stacked generalisation approach, we follow the late fusion scheme described above. Given a sequence of Kinect's data streams, we extract feature vectors from users' facial expressions as well as from body gestures. Then, the extracted feature vectors are fed to separate unimodal classifiers. After learning each NN model, the posteriors of the hidden variables can then be used as a new representation for the data. By adopting unified representations of the data, we can learn high-order correlations across modalities. Deeper networks with fewer hidden variables can provide simpler, more descriptive model. Therefore, we consider training an NN over the pre-trained layers for each modality, as motivated by deep learning methods. We stack NNs and train them layer-wise by starting at the base layer and moving up. This is a directed model since there is no feedback from higher layers to the lower layers, as shown in the lower part of Figure 25. This layer-wise architecture improves performance, while avoiding overfitting.

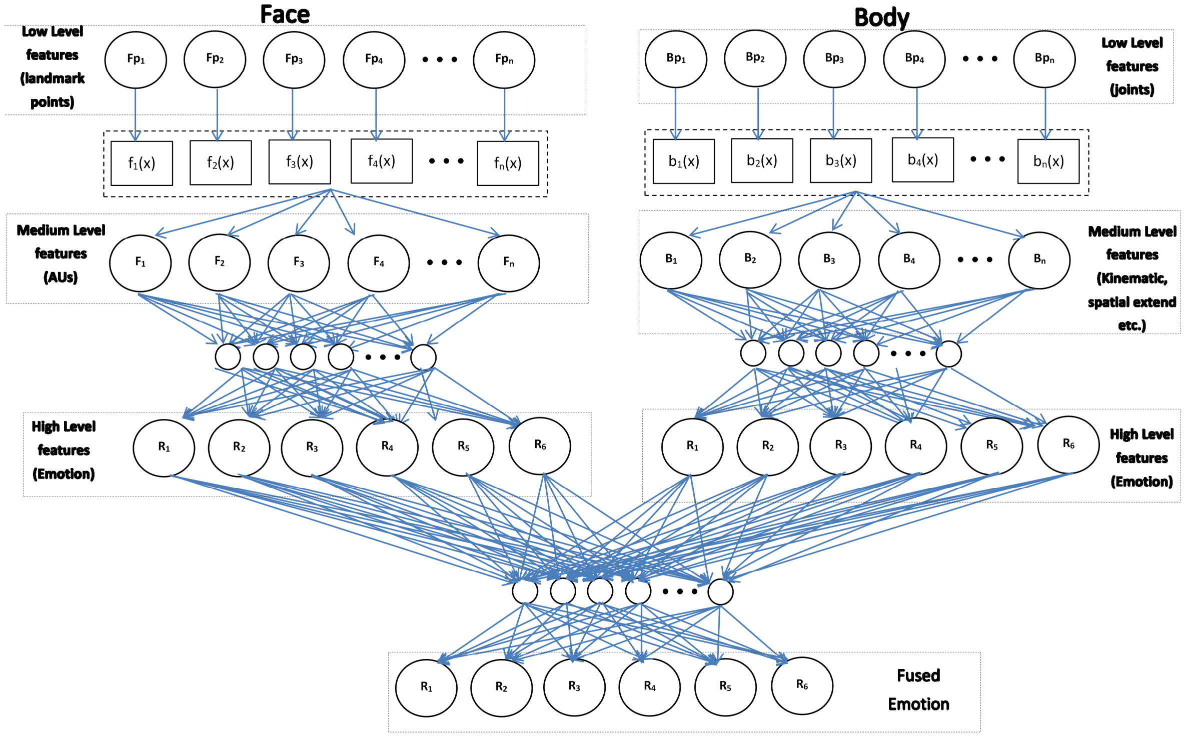


Figure 25: Dynamic fusion architecture

For the evaluation of the proposed multimodal affective state recognition method, we examined whether the proposed fusion algorithm performs better than the intermediate mono-modal classifiers as well as a number of different multimodal approaches. More specifically, we compared the performance of the proposed algorithm against the recognition rates of mono-modal classifiers (both face and body) as well as against the recognition rates of various multimodal schemes (e.g., Linear Weighted, Non Linear SVM and Shallow MLP). As shown in Figure 26, the proposed model outperforms all other methods, both mono-modal and multimodal (early and late fusion approaches), with a recognition rate of 98.3%. As we can see, the two mono-modal classifiers provide high recognition rates similar to those of early fusion algorithms, i.e., non Linear SVM (N-L SVM) and Shallow NN, while the proposed fusion method outperforms the linear weighted-based late fusion approach, with an improvement of 6,7% (Psaltis et al, 2016).

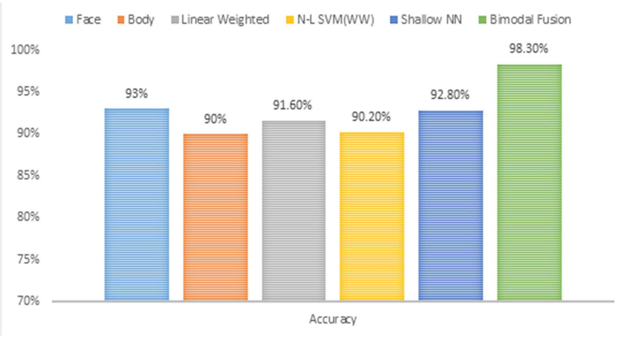


Figure 26 - Comparison of the proposed fusion algorithm against face and body mono-modal classifiers, two early fusion approaches (N-L SVM and Shallow NN) and a late fusion multimodal method (Linear Weighted).

A. Psaltis, K. Kaza, K. Stefanidis, S. Thermos, K. Apostolakis, K. Dimitropoulos, P. Daras, "Multimodal Affective State Recognition in Serious Games Applications", IEEE International Conference on Imaging Systems and Techniques, Chania, Greece, October 4-6, 2016, accepted for publication.

6 Conclusions

This deliverable builds on the work described in D3.2. Whereas the original deliverable focused on an approach to describe prosociality based on membership to one or more core domains this deliverable moves to a skills based approach. This shift is based on improved understanding in both how children acquire skills and specifically prosocial skills but also on observational difficulties for core domain models.

The deliverable covers the various technologies which will be embedded in the ProsocialLearn platform. These technologies are either to do with how to measure the state of a player or how to represent the state.

This deliverable describes concrete solutions to the problems of how to synchronise the remotely distributed devices. Additionally, it covers how to fuse at a feature level and the basis of an approach for decision level fusion. This twostep approach is needed due to the difficulty of building large multi-modal data sets and associated classifiers. The use of decision fusion allows an educated combination of sensor results which is robust to noise and missing data.

The representation of the players state was measured via game interactions (in-game and sensor observations). These were described using the experience API. This common representation allows us to build a non-mutable user profile (or Learning Record Store) which can be used to deliver information of interest to teachers.

Visualisation of game interactions and player state will be covered in the next iteration of this deliverable. Alongside this will be extensions of the various sections presented here.

7 References

Atrey, P.K., Kankanhalli, M.S. & Jain, R., 2006. Information assimilation framework for event detection in multimedia surveillance systems. *ACM Multimedia Systems Journal*, 12(3), pp.239–253.

Attfield, S., 2011. Towards a science of user engagement (position paper). In *Proceedings of WSDM Workshop on User Modelling for Web Applicaitons*.

Bann, E.Y. & Bryson, J.J., 2013. The conceptualisation of emotion qualia: Semantic clustering of emoiton tweets. In *Proceedings of the 13th Neural Computation and Psychology Workshop*.

Banos, R.M., 2004. Immersion and emotion: their impact on the sense of presence. *Cyber-Psychology and Behaviour*, 7(6), pp.734–741.

Biswas, P. & Robinson, P., 2012. A brief survey on user modelling. In *HCI in Speech, Image and Language Processing for Human Computer Interaction*.

Blandford, A., Butterworth, R. & Curzon, P., 2004. Models of interactive systems: a case study on programmable user modelling. *International Journal of Human-Computer Studies*, 60, pp.149–200.

Bradley, M.M. et al., 1992. Remembering pictures: Pleasure and arrousal in memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18(2), pp.379–390.

Bradley, M.M. & Lang, P.J., 1999. *Affectve norms for English words (ANEW): Instruction manual and affective ratings*,

Bredin, H. & Chollet, G., 2007. Audio-visual speech synchrony measure for talking-face identity verification. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*. pp. 233–236.

Bridgeland, J., Bruce, M. & Hariharan, A., 2013. *The missing piece: A national teacher survey on how social and emotional learning can empower children and transform schools*, Civic Books.

Brierley, P. & Batty, B., 1999. Data mining with neural networks: an applied example in understanding electricity consumption patterns. In *Knowledge Discovery and Data Mining*. pp. 240–303.

Broadie, A., 1991. Trust. Presentation given for the Henry Duncan prize.

Brockmyer, J.H., 2009. The development of the Game Engagement Questionnaire: A measure of engagement in video game playing. *Journal of Experimental Social Psychology*, 45(4), pp.624–634.

Brusilovsky, P., 2004. KnowledgeTree: A Distributed Architecture for Adaptive E-Learning. In *International Conference on World Wide Web*. pp. 104–113.

Carmagnola, F., Cena, F. & Gena, C., 2011. User model interoperability: a survey. *User Modelling and User-Adapted Interaction*, 21(3), pp.295–331.

Castillejo, E., Almeida, A. & Lopez-de-Ipina, D., 2014. Modelling users, context and devices for adaptive user interface systems. *International Journal of Pervasive Computing and Communications*, 10(1), pp.69–91.

Chetty, G. & Wagner, M., 2005. Audio-visual multimodal fusion for biometric person authentication and liveness verification. In *Proccedings of the 2005 NICTA-HCSNet Multimodal User Interaction Workshop*. pp. 17–24.

Chieu, H.L. & Lee, Y.K., 2009. Query based event extraction along a timelien. In *Proceedings of the ACM Conferenc eon Research and Development in Information Retrieval*. pp. 425–432.

Choi, S. et al., 2005. Blind Source Separaton and Independepent Component Analysis: A review. *Neural Information Processing - Letters and Reviews*, 6(1), pp.1–57.

Christakis, N.A. & Fowler, J.H., 2009. *Connected: The surprising power of our social networks and how they shape our lives*, Little, Brown and Company.

Clemmensen, T., 2004. Four approaches to user modelling – a qualitative research interview study of HCI professionals' practice. *Interacting with Computers*, 16(4), pp.799–829.

Cocea, M. & Magoulas, G., 2015. Participatory Learner Modelling Design: A methodology for iterative learner models development. *Information Sciences*, 321(10), pp.48–70.

Cowie, R. et al., 1999. What a neural net needs to know about emotion words. In *Proceedings 3rd World Multiconference on Circuits, SYstems, Communications, and Computers*.

Dempster, A., 1968. A generalisation of Bayesian inference. *Journal of the Royal Statistical Society*, pp.205–247.

Dunn, J.R. & Schweitzer, M.E., 2005. Feeling and believing: the influence of emotion on trust. *Journal on Personality and Social Psychology*, 88(5), p.736.

Ekman, P., 1992. An argument for basic emotions. *Cognition and Emotion*, 6, pp.169–200.

Fraser, D.C. & Potter, J.E., 1969. The optimum linear smoother as a combination of two optimum linear filters. *IEEE Transactions on Automation and Control*, 14(4), pp.387–390.

Gan, Q. & Harris, C.J., 2001. Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion. *IEEE Transactions in Aerospace and Electronic Systems*, 37(1), pp.273–279.

Gazor, S., 2003. Speech Probability Distribution. *IEEE Signal Processing Letters*, 10(7), pp.204–207.

Heinzel, G., Rudiger, A. & Shilling, R., 2002. *Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new at-top windows*,

Hua, X.S. & Zhang, H.J., 2004. An attention-based decision fusion scheme for multimedia information retrieval. In *Proceedings of 5th Pacific-Rim Conference on Multimedia*.

Irwin, H.J., 1999. Pathological and non-pathological dissociation: The relevance of childhood trauma. *The Journal of Psychology*, 133(2), pp.157–164.

Iyengar, G., Nock, H.J. & Neti, C., 2003. Audio-visual synchrony for detection of monologue in video archives. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Jaimes, A. & Sebe, N., 2005. Multimodal Human Computer Interaction: A Survey. Computer Vision in Human-Computer Interaction. *Lecture Notes in Computer Science*, 3766, pp.1–15.

Jain, A., Nansakumar, K. & Ross, A., 2005. Score normalization in multimodal biometric systems. *Pattern Recognition*, 38(12), pp.2270–2285.

Kahou, S.E. et al., 2015. EmoNets: Multimodal deep learning approaches for emotion recognition in video. Available at: http://arxiv.org/abs/1503.01800.

Kardan, A., Aziz, M. & Shahpasand, M., 2015. Adaptive systems: a content analysis on technical side for e-learning environments. *Artificial Intelligence Review*, 44(3), pp.365–391.

Keltner, D. et al., 2014. The sociocultural appraisals , values, and emotions (SAVE) framework of prosociality: Core processes from gene to meme. *Annual Review of Psychology*, 65, pp.94–107.

Lehmann, J., 2012. Models of user engagement. In *User Modelling, Adaptation, and Personalization*. pp. 164–175.

Lehmann, L. & Keller, L., 2006. The evolution of cooperation and altruism -- a general framework and a classification of models. *Journal of Evolutionary Biology*, 19(5), pp.1365–1376.

Luhmann, N., 2000. Familiarity, confidence, trust: Problems and alternatives. *Trust: Making and breaking cooperative relations*, 6, pp.94–107.

Marsh, S.P., 1994a. *Formalising trust as a computational concept*,

Marsh, S.P., 1994b. Optimisim and pessimism in trust. In *Proceedings of Ibero-American Conference on Artificial Intelligence*.

Martinez-Villasenor, M. de L., 2014. Enrichment of Learner Profile with Ubiquitous User Model Interoperability. *Computacion y Sistemas*, 18(2), pp.359–374.

McGinnis, E. & Goldstein, A.P., 1997. *Skillstreaming the elementary school child: New strategies and perspectives for teaching prosocial skills*, Research Press.

Mehrabian, A., 1996. Pleasure-Arousal-Dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology: Developmental, Learning, Personality, Social*, 14, pp.261–292.

Moneta, G.B. & Mihaly Csikszentmihalyi, 1999. Models of concertation in natural environments: A comparitive approach based on streams of experimental data. *Social Behaviour and Personality*, 27(6), pp.603–637.

Morrison, D., Wang, R. & De Silva, L.C., 2007. Ensemble methods for spoken emotion recognition in call-centres. *Speech Communication*, 49(2), pp.98–112.

Mroueh, Y., Marcheret, E. & Goel, V., 2015. Deep Multimodal Learning for Audio-Visual Speech Recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Newell, A. & Simon, H.A., 1995. GPS, a program that simulates human thought. In *Computers & though*. pp. 279–293.

Nowak, M.A. & Roch, S., 2007. Upstream reciprocity and the evolution of gratitude. *Proceedings of the Royal Society B (Biological Sciences)*, 274(1610).

O'Brien, H.L. & Toms, E.G., 2008. What is user engagement? A conceptual framework for defining user engagement with technology. *Journal of the American Society for Information Science and*

Technology, 69(6), pp.938–955.

Olsen, T., Procci, K. & Bowers, C., 2011. Serious games usability testing: How to ensure propoer usability, playability, and effectiveness. In *Design, user experience, and usability: theory, methods, tools, and practice*. pp. 625–634.

Payne, S.J. & Green, T.R.G., 1986. Task-Action Grammars: A Model of Mental Representation of Task Languages. *Human Computer Interaction*, 2(2), pp.93–133.

Peleg, B. & Sudholter, P., 2005. Introduction to the theory of cooperative games. *Games and Economic Behaviour*, 53(2), pp.269–270.

Plutchik, R., 2001. The nature of emotions. *American Scientist*, 89(4).

Rashidi, A. & Ghassemian, H., 2003. Extended dempster–shafer theory for multi-system/sensor decision fusion. In *Proceedings of Commission IV Joint Workshop on Challenges in Geospatial Analysis*. pp. 31–37.

Resnick, H. & Sherer, M., 1994. Electronic tools for social work practise and exucation: Part 1. *Computers in Human Services*, 11(1), p.2.

Rickard, S., 2007. The DUET blind source separation algorithm. In *Blind Source Separation*. pp. 217–241.

Rotenberg, K.J., 2010. The conceptualisation of interpersonal trust: A basis, domain, and target framework. *Interpersonal trust during childhood and adolesence*, pp.8–27.

Rotenberg, K.J., Boulton, M.J. & Fox, C.L., 2005. Cross-sectional and longitudinal relations among children's trust beliefs, psychological maladjustment, and social relationships: are very high as well as very low trusting children at risk? *Journal of Abnormal Child Psychology*, 33(5), pp.595–610.

Russell, J.A., 1980. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6), pp.1161–1178. Available at: http://content.apa.org/journals/psp/39/6/1161.

S. D. Boon & J. G. Holmes, 1991. The dynamics of interpersonal trust: Resolving uncertainty in the face of risk. *Cooperation and Prosocial Behaviour*, p.190.

Sanderson, C. & Paliwal, K.K., 2004. Identity verification using speech and face information. *Digital Signal Processing*, 14(5), pp.229–280.

Schroff, F., Kalenichenko, D. & Philbin, J., 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Available at: http://arxiv.org/abs/1503.03832.

Shafer, G., 1976. *A mathematical theory of evidence*,

Singer, M.J. & Witmer, B.G., 1999. On selecting the right yardstick. *Presence: Teleoperators and Virtual Environments*, 8(5), pp.566–573.

Snoek, C.G.M., Worring, M. & Smeulders, A.W.M., 2005. Early versus late fusion in semantic video analysis. In *Proccedings of the 13th annual ACM International Conference on Multimedia*.

Tamborini, R. & Skalski, P., 2006. *The role of presence in the experience of electronic games*, Terusaki, K. & Stigliani, V., 2014. *Emotion Detection using Deep Belief Networks*,

Wang, J. et al., 2003. Experiential sampling for video surveillance. In *Proceedings of ACM Workshop on Video Surveillance*.

Wang, X. & Zhao, T., 2014. Operation condition monitoring using temporal weighted Dempster-Shafer theory. In *Annual Conference of the Prognostics and Health Management Society*.

Wang, X.H. et al., 2004. Ontology based context modelling and reasoning using OWL. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshop*. pp. 18–22.

Whitehill, J., 2014. The faces of engagement: Automatic recognition of student engagement from facial expressions. *IEEE Transactions on Affective Computing*, 5(1), pp.86–98.

Wu, H. et al., 2002. Sensor fusion using Dempster-Shafer theory. In *Proccedings of the IEEE Instrumentation and Measurement Conference*.

Yan, R., Yang, J. & Hauptmann, A., 2004. Learning query-class dependent weights in automatic video retrieval. In *Proceedings of ACM International Conference on Multimedia*. pp. 548–555.

Zins, J.E. et al., 2004. The scientific base linking social and emotional learning to school success. In *Building academic success on social and emotional learning: What does the research say?*. pp. 3–22.

Appendix A: User modelling state of the art

Efforts to model the users of interactive systems has its origins in the early work of computer scientists and psychologists during which the development of specification models of human cognition to predict task performance at the user interface (Biswas & Robinson 2012) was carried out. Notable examples of this pioneering work include the introduction of formal grammars to model user interactions (Payne & Green 1986; Newell & Simon 1995). This progressed into the construction of reasoning frameworks that could be tailored for specific application domains (Blandford et al. 2004). With the advent of wide spread use of mobile and ubiquitous interactive devices and sensors (Jaimes & Sebe 2005), the breadth and depth of contextual information and influences impacting human-computer interaction expanded significantly (Castillejo et al. 2014). Concomitant with the growth of mobile computing, the wide spread engagement with online communities through social networking platforms generated a demand for flexible approaches to modelling users interactions through the application of ontology based representation systems (Wang et al. 2004).

Within the large corpus of research relating to player modelling is the *learner model* sub-domain in which information about the learner such as subject knowledge; learning preferences and goals; user background and traits; and contextual state are considered. It is typical that the information aggregated to represent users is collected from a variety of data sources (Brusilovsky 2004) – various interoperability efforts exist to integrate these (Martinez-Villasenor 2014; Carmagnola et al. 2011). In (Kardan et al. 2015), Kardan et al. reviews a number of approaches to using such data in adaptive processes that assist learning process in conventional learning domains. These include identifying student preferences and learning style using machine learning techniques; ontological approaches to representing user knowledge; and learner management software suites. In their review, systems and methodologies that address aspects of *social learning* are identified as a new and emerging sub-discipline of the field.

A methodology for the design and development of learner models has been described by (Cocea & Magoulas 2015) as process in which the following questions are asked:

- What is being modelled?
- How is the information represented?
- How is the model maintained?

These questions form the basis of a requirements analysis that will shape the architecture and functionality of the user model. In order to answer these questions an iterative, multi-stage design process takes place. A preliminary conceptual user model (based on the knowledge domain) is proposed first. Following this, related data is collected from test scenarios in which system adaptation is expected to play a role. Next, a mapping of the collected data to the conceptual model is attempted and evaluation of its efficacy in characterising user behaviour is carried out - the results of the evaluation feed into the next cycle of design (see Figure 27).

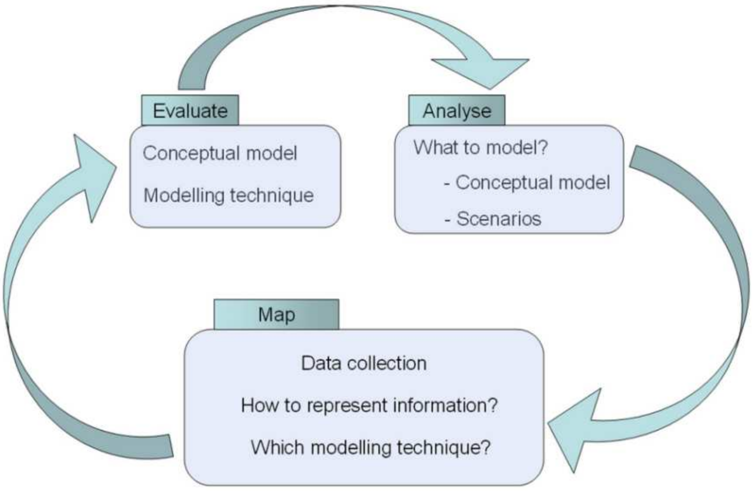


Figure 27: Cocea et al.'s Player modelling design methodology

The primary role of player modelling in the ProsocialLearn project is to effectively represent the student learners with information that allow games to adapt based on their interactive and affective responses during game play – an application of player modelling that has been identified as a novel and emerging field of research. For this reason we will be guided by the player modelling design methodology provided by (Cocea & Magoulas 2015); in this document we begin this process by setting out our initial conceptual model in terms of what is being modelled; specifying how the information will be represented and describing its maintenance and persistence. In the sections that follow we describe what is being modelled; how the data is represented and how it is maintained.

Appendix B: Multimodal fusion state of the art

In this section, we provide an overview of the different fusion methods have been used in the literature to perform various multimedia analysis tasks. The fusion methods are divided into the following categories: statistical rule-based methods, classification based methods (see Table 11).

Multimodal Fusion Methods	
Statistical Rule based	**Classification-based**
Linear Weighted Fusion,	SVM,
Majority Voting	Naïve Bayes,
	Neural Networks

Table 11: A list of multimodal fusion methods per category

Statistical rule-based fusion methods

The rule-based fusion method includes a variety of basic rules of combining multimodal information. These include statistical rule-based methods such as linear weighted fusion (sum and product), MAX, MIN, AND, OR, majority voting. The rule-based schemes generally perform well if the quality of temporal alignment between different modalities is good. In the literature has been used for face detection, human tracking, monologue detection, speech and speaker recognition, image and video retrieval, and person identification.

- **Linear weighted fusion**

Linear weighted fusion is one of the simplest and most widely used methods. In this method, the information obtained from different modalities is combined in a linear fashion. The information could be the low-level features (e.g. pixel positions), mid-level characteristics (e.g. distances) or the semantic-level emotional states (e.g. happy, sad). To combine the information, we need to assign normalized weights to different modalities. Researchers used computational and estimation methods to normalize the different modality weights (Jain et al. 2005). Features or decision could be fused using sum or dot operators:

$$\sum_{i=1}^{n} w_i \times l_i \qquad\qquad \prod_{i=1}^{n} {l_i}^{w_i}$$

Several researchers have adopted the linear fusion strategy both at the feature level (Yan et al. 2004; Wang et al. 2003; Iyengar et al. 2003), and decision level (Hua & Zhang 2004; Yan et al. 2004) for performing various multimedia analysis tasks. *Majority voting* is a special case of weighted combination with all weights to be equal. In majority voting based fusion, the final decision is the one where the majority of the classifiers reach a similar decision (Sanderson & Paliwal 2004).

This method is computationally less expensive compared to other methods. However, it is observed that the optimal weight assignment is the major drawback of the linear weighted fusion method. The issue of determining and adjusting the weights for different modalities is an open research issue.

Classification-based fusion methods

Instead of naively combining the data using statistic rule-based methods, it is possible to use classification techniques that have been used to classify the multimodal observation into one of the pre-defined classes. The methods in this category are the support vector machine (SVM), Naïve Bayes and Neural Networks.

- **SVM**

One of the most common approaches is to employ SVM. SVM has become increasingly popular for data classification and related tasks. It has been used by most researchers in tasks including feature categorization, concept classification, face detection, and modality fusion. From the perspective of multimodal fusion, SVM is used to solve a pattern classification problem, where the input to SVMs' classifier is the decision scores given by the unimodal classifier.

(Bredin & Chollet 2007) used a discriminate learning approach while fusing different modalities at the semantic level. For example, in Figure 28 the decision scores (probabilistic output) of all intermediate concept classifiers are used to construct a semantic feature vector that is passed to the final decision layer in SVM.

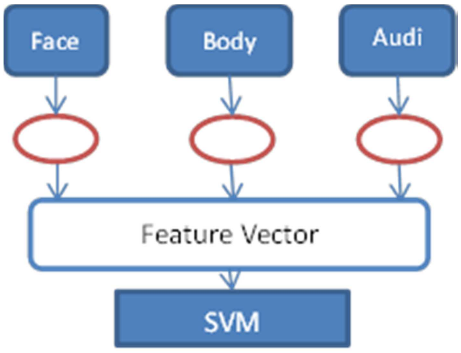


Figure 28: Support Vector Machine based fusion

- **Naïve Bayes**

This approach is very simple to apply to data at feature as well as at decision level. The Bayesian inference is often referred to as the 'classical' method for fusion multimodal data acquired by various sensors. It has been used very commonly in fusion problems since it has been the basis for many other methods. The observations obtained from multiple modalities or the decisions obtained from different classifiers are combined, and an inference of the joint probability of an observation or a decision is derived (Rashidi & Ghassemian 2003).

For statistically independent modalities, the joint probability of a hypothesis H based on the fused decisions can be computed as:

$$p(H|D_1, D_2, .., D_n) = \frac{1}{N}\prod_{k=1}^{n} p(D_k|H)^{w_k}$$

This posterior probability is computed for all possible hypotheses E. The hypothesis that returns the maximum probability is determined using the MAP rule:

$$\underset{H\in E}{\operatorname{argmax}}\, p(H|D_1, D_2, .., D_n)$$

One of the major advantages of Bayesian inference is that it can compute the posterior probability of the hypothesis based on the new observations. It requires a priori and the conditional probabilities of the hypothesis to be well defined. In absence of any knowledge of suitable priors, the method does not perform well. Dempster-Shafer theory comes as a solution to this as it allows defining the priors needed.

- **Neural Networks**

Like other machine learning methods, Neural network (NN) have been used to solve a wide variety of tasks including multimodal fusion. Neural networks are considered a non-linear black box that is used to estimate functions that can depend on a large number of input data and are generally unknown. Basically the network 'learns' from the observed data to recognize patterns and produce noiseless outputs. In addition it has ability to generalize, as it produces outputs for inputs it has not been taught how to deal with (unseen data).

The output of a neuron is a function of the weighted sum of the inputs plus a bias term:

$$f(I_1 w_1 + I_2 w_2 + \ldots + I_n w_n + bias)$$

The network architecture design between the input and output nodes is an important factor for the success or failure of this method (see Figure 29). The weights along the paths, that connect the input nodes to the output nodes, decide the input–output mapping behaviour. These weights can be adjusted during the training phase to obtain the optimal fusion results. The most common technique is to adjust the weights so that the difference between the network output *"predicted"* and the required output *"ground truth"* is reduced Brierley et al back propagation method (Brierley & Batty 1999)

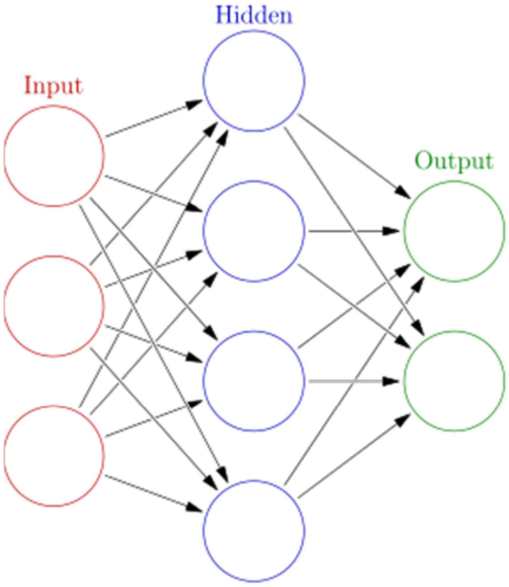


Figure 29: Artificial Neural Network

- **Deep Learning**

Deep Neural Networks (Neural Networks with more than one hidden layer) have become increasingly popular. Unlike the classical ANN, a pyramid of artificial neurons split into several layers, where each layer takes input data from the layer below. It has been widely used to convert large amount of data, in most cases noisy data, into smaller amount of better structure info. We can train deep networks to produce useful representations. Deep learning solutions are very powerful, they are the state of the art in several machine learning problems (Rashidi & Ghassemian 2003; Mroueh et al. 2015; Kahou et al. 2015; Terusaki & Stigliani 2014; Schroff et al. 2015). Figure 30 illustrates Deep Learning network architecture.

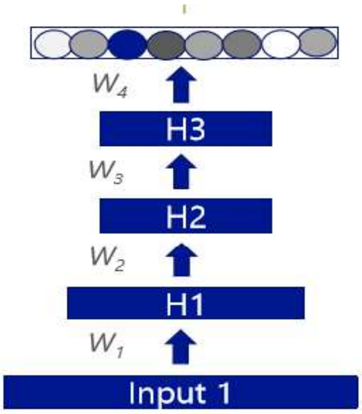


Figure 30: Typical deep learning architecture

Deep learning networks can be applied at feature level as well as at decision level, being trained directly on raw data or decisions accordingly. The layer-wise architecture improves performance while avoiding overfitting. Deeper networks with fewer hidden variables can provide simpler, more descriptive model. However, these networks are hard to optimize, and the more layers they include the longer time it takes to be trained. Figure 31 presents a deep network which fuses visual cues at the first layer utilizing the correlation between multimodal features at an early stage, while adding more informative features about the state of the game in a latter fusion layer. Fused information could be used as a measure of confidence for user engagement.

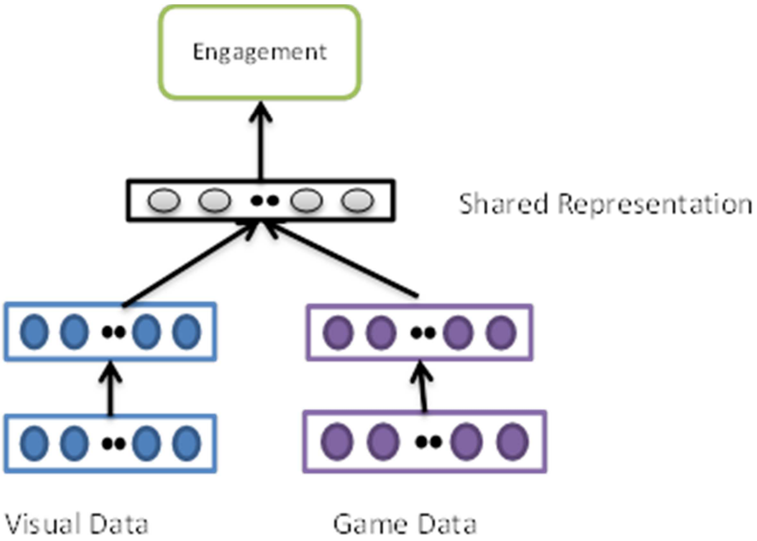


Figure 31: Fusion using a deep neural network

Fusion approaches

- **Feature Fusion**

According to literature, the most widely used strategy is to fuse the data at the feature level (also called early fusion). Snoek et al., defines early fusion as “the fusion scheme that integrates unimodal features before learning concepts” (Snoek et al. 2005). In the early fusion approach, the monomodal features are first extracted. After analysis of monomodal signals, the extracted features from each modality are combined into a single representation. A simple technique would be to concatenate all features from multiple cues into one feature vector (multimodal representation). Using machine learning approaches, we can train a classifier with few examples to learn higher semantic concepts. Figure 32 shows a bimodal early fusion scheme, where initially data fused and then fed to the classifier.

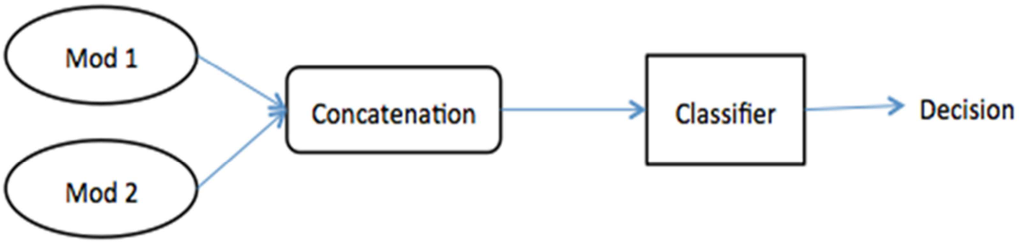


Figure 32: Bi-modal early fusion scheme

In the feature level fusion approach, the number of features extracted from different modalities may be numerous. In the following, we present the features of each modality, which give us an indication of the user's engagement, attention.

- **Visual features**: According to Deliverable D3.1 it may include features based on eyes (e.g. *Distances between eyes' and eyelids.*), eyebrows (e.g. *Angles between eyes and eyebrow)* mouth (e.g. *Distances between mouth and lips*, gaze (e.g. Gaze Distance, Location on screen, Pupil diameter, Blinking), and so on.
- **Motion features**: Motion can be represented in the form of kinetic energy which measures the pixel variation within a shot, motion direction and magnitude histogram, optical flows and motion patterns in specific directions. Motion vector includes features based on body (e.g. *Kinetic energy, contraction index, density, smoothness, fluidity, symmetry, forwards/backwards leaning of the upper body and relative positions, directness), head (e.g. yaw, pitch roll of the head), hand (e.g. velocity acceleration, fluidity of hand barycenter).* These features are extracted through joint-oriented skeleton tracking using depth and RGB information from Kinect sensor. Hand features are gathered for both hands using Kinect sensor in a full-body movement tracking environment or for a single hand using LEAP Motion sensor.
- **Audio features:** The audio features may be generated based on the short time Fourier transform including the fast Fourier transform (FFT), mel-frequency cepstral coefficient (MFCC) described in Deliverable D3.1.
- **Text features:** The textual features can be extracted from chat messages in the context of Negative, Neutral and Positive.
- **Metadata**: The metadata features are used as supplementary information in the production process, such as the game event, the time stamp as well as the duration and effect of the action performed. They can provide extra information to audio or visual features. The context is accessory information that greatly influences the performance of a fusion process.

There are several advantages of fusing the modalities at feature level. The multimodal feature representation might be the most important. Since the features are integrated from the beginning of the process, the fusion utilizes the correlation between multimodal features at an early stage. In addition, the requirement of only one learning phase leads to better performance. However, it is hard to combine and synchronize all these multimodal features into a common representation because of their different format and processing time.

- **Decision level Fusion**

The other approach is decision level fusion or late fusion which fuses multiple modalities in the semantic space. In a similar way to early fusion, Snoek et al., defines late fusion as the "fusion scheme that first reduces unimodal features to separately learned concept scores, and then these scores are integrated to learn concepts". In the late fusion approach, the monomodal features are first extracted. After analysis of monomodal signals, the extracted features from each modality are fed to a modality specific classifier described in D3.1. Each classifier is trained to provide a local decision. The local decisions are then combined into a single semantic representation, which further analysed to provide the final decision about the task. In terms of engagement, local classifiers return a confidence as a probability in the range of [0, 1]. A bi-modal fusion scheme for late fusion is illustrated in Figure 33.

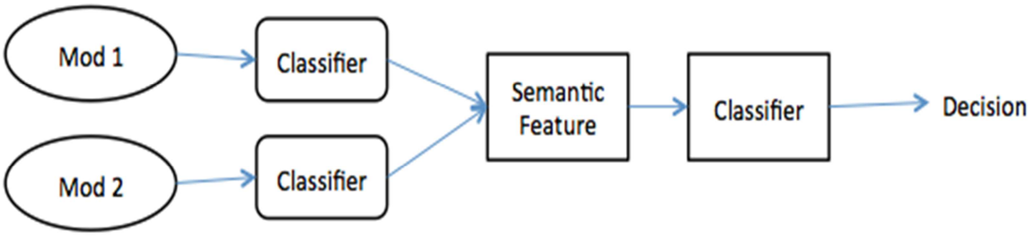


Figure 33: bi-modal late fusion scheme

Early and Late fusion approaches mainly differ in the way they combine the results from feature extraction on the various modalities. The latter fuses unimodal decisions into a multimodal semantic representation rather than a multimodal feature representation. As a result, the fusion of decisions becomes easier, while reflecting the individual strength of modalities. Moreover, the late fusion approaches are able to draw a conclusion even when some modalities are not presented in the fusion process, which is hard to achieve in the early fusion approach. In addition, late fusion schemes offer flexibility, in a way that different analysis models could be used to different modalities (e.g. SVM for face features, ANN for body features). In contrast with early fusion techniques, decision level approaches fail to utilize the feature level correlation among modalities. Furthermore, as every modality requires different classifier to obtain the local decisions, the learning process becomes quite expensive and hinders the overall performance.

- **Examples of decision level fusion**

In this section we will examine the use of decision level fusion to improve the results of multimodal fusion. Decision level fusion operates on the outputs of classifiers. It will not perform well in the cases where all the fused classifiers are performing poorly. We will examine a number of different approaches: naïve, using statistical smoothing, and Dempster-Shafer evidence theory.

The first step is to create datasets to be fused. We can achieve this using the results from the classifier run over the test set (as described in D3.1) and a synthetic dataset. To make the comparison simpler we will just use a simulated measure of emotion that comes from something in the game. Both of the emotions measures return a confidence in the range of [0,1]. In terms of the voice emotion classifiers this has required that the classifiers be trained to produce a probabilistic output. The graph in Figure 34 illustrates the waveforms along with the ground truth value. Examination of the figure shows the estimates of emotion to be relatively uncorrelated. The other emotion detector is relatively better at finding the cases where the classifier finds no emotion compared with the voice emotion signal. The root mean square (RMS) error of the sequence is shown in Table 12. The other emotion signal has roughly twice the error of the voice emotion.

Signal	RMS-error
Voice emotion	0.112181088435
Other emotion	0.236679782547

Table 12: RMS error for the different signals

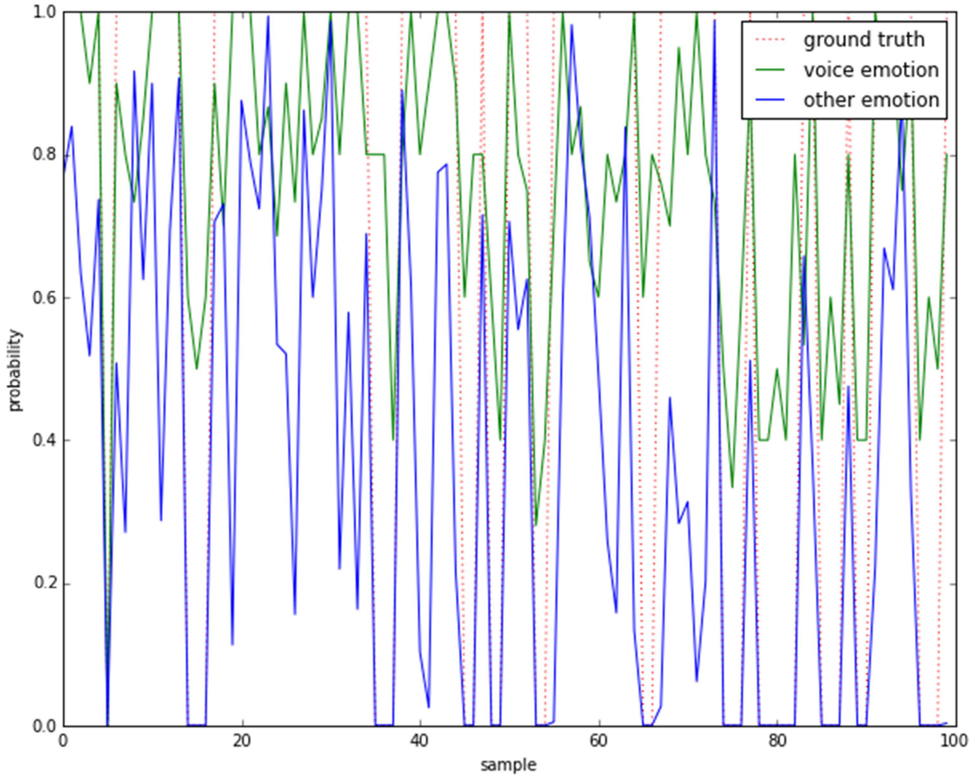


Figure 34: Emotional signals for input to the fusion system along with ground truth

- **Naïve Fusion**

The simplest fusion approach is to use the most common results for the classifiers. This uses a simple statistical metric. There are a number of choices here. The most common of which are the mean, mode, and median of the data. In the case of an unknown distribution the median is a better descriptor as it is more robust. As an example if the distribution is very skewed the median is the best measure. In the case of a normally distributed signal then median, mean and mode behave similarly. The result of performing median based fusion is shown in Figure 35. In reality we will not use the signal like this but perform a hard threshold on the data. The RMS errors are shown in Table 13. In both cases there is a definite improvement over the raw signals.

Signal	RMS-error
median	0.0992
Thresholded median	0.0700

Table 13: RMS errors of the fused signals

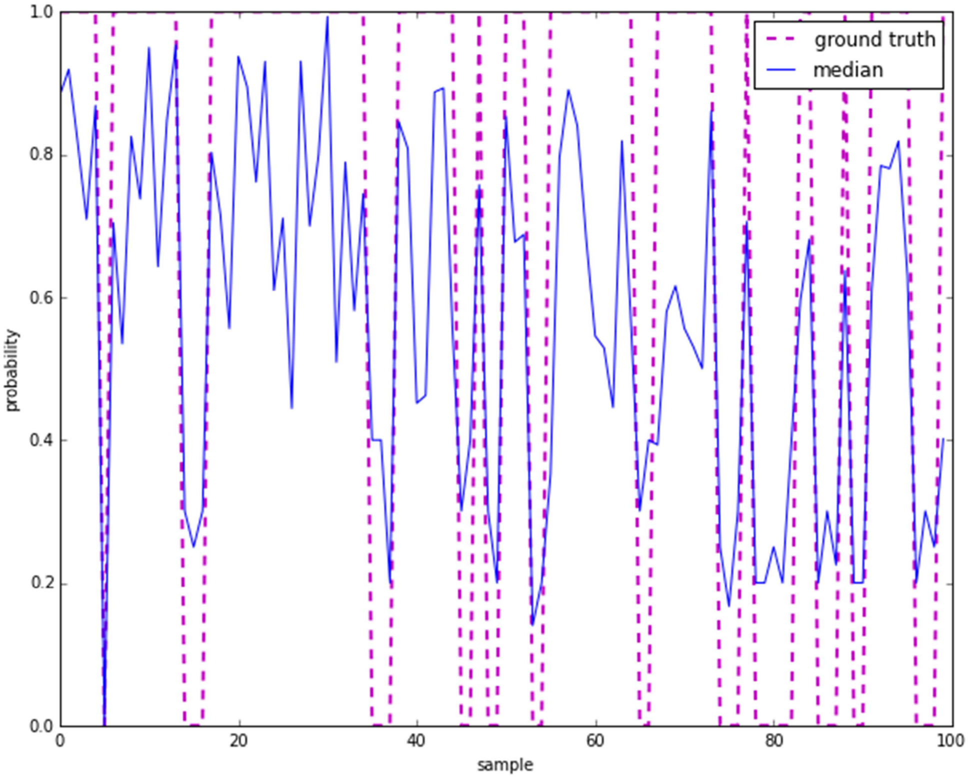


Figure 35: Fusion based on median of the values

While this approach shows an improvement over the original signals it is a very noisy signal. This noise is especially prevalent in the region around 0.5 which is the decision threshold. A consequence of this is that it is very likely that the state will get flipped erroneously.

- **Statistical Smoothing based Fusion**

Instead of naively combining the data using a summary statistic it is possible to create a weighted sum of the different signals. One of the most common approaches is to employ Kalman filters (Gan & Harris 2001). However, this requires a sufficient model of the various sensors and their states. For information that is coming via the game this is not always an option. The approach we employ is based on statistical smoothing. Specifically we use Fraser-Potter smoothing (Fraser & Potter 1969). The essence of this approach is to weight each signal by the inverse of the statistical variance and normalise the result:

$$x_3 = \frac{1}{\sigma_1^2 + \sigma_2^2}\left(\frac{x_1}{\sigma_1^2} + \frac{x_2}{\sigma_2^2}\right)$$

This approach is very simple to apply to data as it only requires the variances to be computed. These can be computed either incrementally or globally (if you know the underlying distribution). The result of applying this weighting to the data gives the result shown in Figure 36. Visually, this result performs similarly to the previous result though with less noise. The RMS errors are given in Table 14. The performance is improved over the naïve case. This supports our intuition that the signal was less noisy.

Signal	RMS-error
Fusion via internal smoothing	0.0910
Thresholded fusion	0.0500

Table 14: RMS errors for fusion via fixed internal smoothing

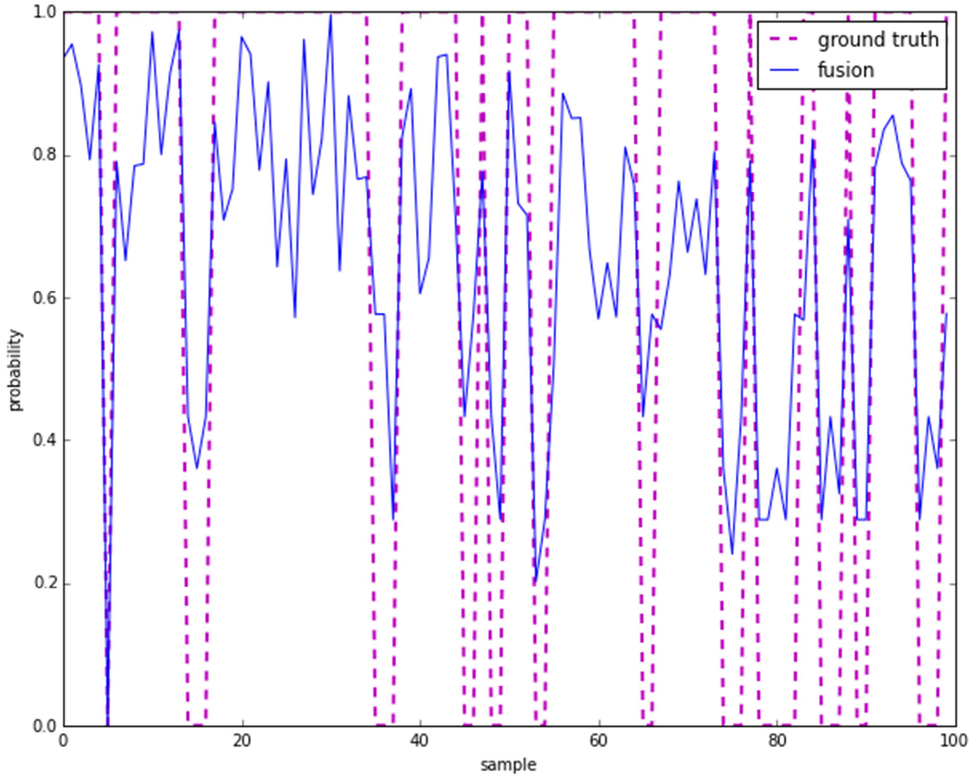


Figure 36: Fusion using a fixed interval smoother

This approach is a significant improvement over the naïve fusion approach. The noise is smaller and the signal is generally smoother. As this has very simple computation and achieves good performance this is a good candidate for the feature level fusion.

- **Dempster-Shafer Fusion**

Dempster-Shafer theory is an extension of traditional probabilistic modelling to allow you to reason over uncertainty (Dempster 1968; Shafer 1976). It is used very commonly in fusion problems as it allows you to relax the requirements to define the priors as needed in Bayesian networks (Wu et al. 2002). Before applying the theory you need to define the frame of discernment. This completely specifies a set in which all the sensors operate. In the case of our system described here the frame of discernment is simple:

$$\Omega = \{NE, E\}$$

From this the power set is formed which is the space over which we reason. The power set includes all the possible combinations of the system:

$$2^{\Omega} = \{\, \phi, NE, E, \{NE, E\}\}$$

At the start of the system, probabilities are assigned to each of these states. The probability, called a mass function, operates on each of the states in the power set. Fusion proceeds via the Dempster-Shafer combination rule:

$$m(A) = m_1 \oplus m_2 = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \phi} m_1(B) m_2(C)}$$

For each sample the evidence is combined using the rule and the fused result is found. The results after doing this are shown in Figure 37. This approach looks even more like the ground truth than the previous approach. The RMS errors are presented in Table 15. As suspected from the visual inspection the performance of this approach is better than the other two fusion techniques.

Signal	RMS-error
DS fusion	0.0657
Threshold of DS Fusion	0.0700

Table 15: RMS error for Dempster-Shafer fusion

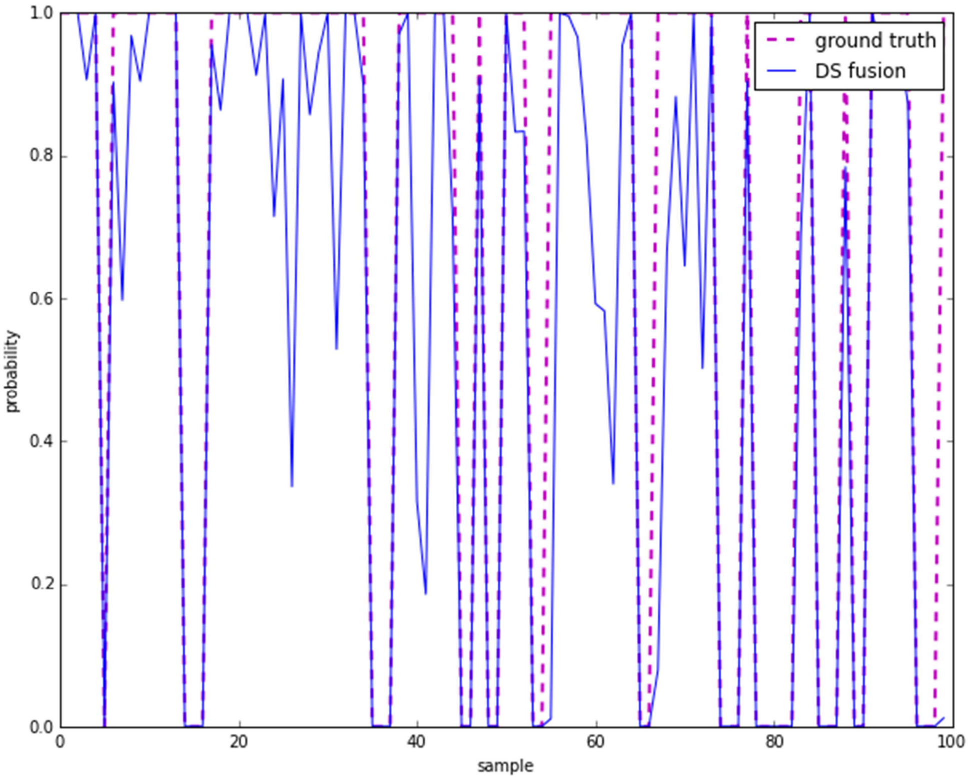


Figure 37: Fusion using Dempster-Shafer evidence theory

It is possible to extend the basic Dempster-Shafer approach in a number of ways. Currently, we assume that each sample is independent of the previous ones. However, as this is a time varying signal this causes some transitions that are erroneous. It would be good to employ a temporal filter over this fusion approach such as described in (Wang & Zhao 2014). This uses an exponential window which allows for identification for trends in the signal. This will be effective here as emotion is unlikely to change radically for a single sample as illustrated by Figure 37. This will be a focus for future work on this approach.

The extra work to move to Dempster-Shafer fusion over statistical smoothing is mostly upfront (in computation of the frame of discernment). The improved performance, along with little computational overhead, indicates that this is a very good approach to apply in the ProsocialLearn platform.

Appendix C: Prosocial Core Domain Models

The SAVE model Socio-cultural Appraisals, Values and Emotions (Keltner et al. 2014) is used for characterizing the propensity to act in a prosocial fashion. It compares costs versus benefit and derives the willingness to act if the benefits are greater than the costs.

The actual formula is presented in Eq.(1):

$$\mathrm{M} \times (\mathrm{D} \times (1 + \mathrm{B}_{\mathrm{self}}) + \mathrm{K} \times \mathrm{B}_{\mathrm{recipient}} - \mathrm{C}_{\mathrm{innaction}}) > \mathrm{C}_{\mathrm{action}} \qquad (1)$$

where:

- M is defined as the social momentum for acting prosocially, or the influence of the socio-cultural milieu. A value for M ranging from 0 to 1 shows social resistance. A value from 1 to infinity shows a positive influence of the milieu. A value of 1 corresponds to an absence of social influence.
- D is defined as the set of individual differences in prosociality and situational factors.
- $\mathrm{B}_{\mathrm{self}}$ is the perceived benefit to oneself for acting prosocially. The benefits can be indefinite.
- K is the set of the giver's biases and perceptions of the specific recipient, which range from positively valenced preferences (e.g., in-group members) to negative values that reflect adversarial stances toward others (e.g., competition, intergroup biases).
- $\mathrm{B}_{\mathrm{recipient}}$ is defined as the benefit another person can receive from the prosocial action (e.g money, friends etc).
- $\mathrm{C}_{\mathrm{innaction}}$ is defined as the cost, or perceived consequences of not acting prosocially. This can take the form of guilt for the individual, or reputation loss, gossip etc at the group level.
- $\mathrm{C}_{\mathrm{action}}$ is the perceived cost to oneself for acting prosocially. For example, prosocial behavior can involve the giving up of a valued resource (e.g., money) to benefit another

It is immediately clear that the above formula, that remains valid in general, needs to be adapted to the case of ProsocialLearn. In particular all the constants in the formula have not yet been firmly determined, still lacking experiments for their calculation. The scenario of ProsocialLearn also relies on the use of on-line games and user observations for the production of information indicating prosociality attitude. It is therefore important to design an operational, possibly simplified, version of the SAVE model that needs to be specialised for a given core domain and akes use of game and observation data.

ProsocialLearn is also trying to include in the assessment of ProSociality the role of emotions as derived from observation. In the SAVE formula emotions are not explicitly considered and it is desirable to overcome such limitation in the used operational models where the emotions would play a direct and explicit role.

The definition of operational model for complex and intrinsically not clear-cut concepts is quite difficult and the work is still in the early stages. This section reports the results achieved so far and considers only the domains of Trust, Cooperation and Fairness. The models present different level of maturity and all of them will evolve and be presented again in the next version of the current deliverable. The other core domains defined in D2.1 will be addressed in the next version of this deliverable.

Trust

Dunn and Schweitzer define trust as “the willingness to accept vulnerability based upon positive expectations about another’s behavior” (Dunn & Schweitzer 2005). In order to formalize a suitable computational model that is suited for prosocial games targeting children aged 7-10, we discriminate between trust factors using Rotenberg’s BDT approach (Rotenberg 2010). BDT, which was explored as part of D2.1, stands for Basis, Domain, Target, and is a 3-dimensional framework of trust. The BDT is defined as follows:

There are three distinct Bases of trust:

1. Honesty, as a base defines when a person is telling the truth and has genuine and not manipulative intent.
2. Emotional Trust, as a base refers to a person refraining from causing emotional harm, such as the case of maintaining confidentiality to disclosures (keeping secrets), and avoiding acts that elicit embarrassment.
3. Reliability, as a base refers to whether a person fulfils his/her word or promises.

Additionally, the three Domains of Trust are defined as follows:

1. The Cognitive/Affective domain refers to a person’s belief that others demonstrate the three Bases of Trust.
2. The Behavior-dependent domain comprises when a person behaviorally expects/relies on others to act in a trusting fashion as per the three Bases of Trust.
3. The Behavior-enacting domain, also referred to as trustworthiness, comprises when an individual behaviorally engages in the three Bases of Trust.

Finally, the framework includes two Target components, which are:

1. Specificity or whether the target of trust is defined as a general category or a specific person.
2. Familiarity or whether the target of trust is defined as slightly or highly familiar.

The BDT framework, therefore, explains that trust includes a defined set of beliefs (expectations) about persons – reliability, emotional trust and honesty – which comprises, at the trusting end of the continuum, positive expectations of their behavior. The framework is graphically depicted in Figure 38.

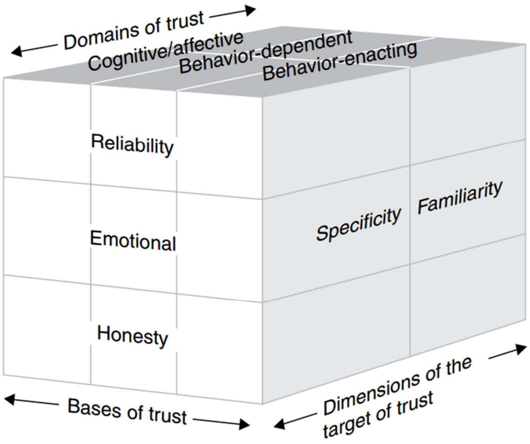


Figure 38: BDT framework structure.

In formalizing a computational model ,we choose to represent trust T as a continuous real variable ($T \in \mathbb{R}$) defined in a specific range: $T \in [-1, +1)$ for each of the three Bases of Trust (honesty, emotional trust, reliability), in a similar manner as the singular trust-value approach by (Marsh 1994a). A value close to -1 basically means that the person has complete distrust in that particular base, therefore demonstrating cynical behaviour. Accordingly, too much trust (blind trust) demonstrated by a value close to $+1$ can be explained as the person being naïve[10]. Both behaviours (i.e., being cynical or being naïve) have been reported to have negative consequences in children social inclusion and academic achievement (Rotenberg et al. 2005). Any form of trust training should focus on children achieving and maintaining moderate levels of trust. Naïve students for example, should be trained in the cognitive/affective (i.e. not to be so quick to believe others demonstrate the three bases of trust) and behaviour-dependent domains (i.e. not to expect others to act trustworthy so blatantly) while cynical students should undergo similar training scenarios in both domains, but tailored with rather opposite goals (like, learning to depend on those who have been proven reliable over the course of time). From this formalization, it becomes apparent that the three bases of trust constitute the theme or scenario of the training episode, i.e. an episode dedicated to honesty, emotional trust or reliability, while the three domains regulate the means by which the scenario is resolved (i.e. whether the child will be called upon to resolve a situation by reconsidering their trust beliefs, expect specific outcomes from other characters and demonstrate trustworthiness themselves). The target component then selects the actors/characters to perform the scenario and deliver the intended training.

- **Computational Model**

We will consider the heuristic formalism of (Marsh 1994a) to model all components of the BDT framework for Trust. In this formalism, developed for use in Distributed Artificial Intelligence (DAI) systems of cooperating autonomous agents, distinctions are made between agents $(a, b, c, \dots)$ and situations $(\alpha, \beta, \gamma, \dots)$. More specifically, situations are modelled in perspective, i.e. from the point of

[10] According to the definition of T, a value of -1 (complete distrust) is possible while a value of $+1$ (blind trust) is not. This is apparent in the definition of trust according to (Broadie 1991), i.e. that it implies a consideration on the part of the trusting party of something or someone in order to search for evidence to believe (in) something and arrive at the decision to trust or not to trust. Complete distrust (-1) complies to this, it implies that, after consideration has occurred, that specific person can definitely not be trusted. A value of $+1$ means that however is being trusted is not being considered at all (why bother, if trusting blindly anyway?). Therefore, blind trust is not trust, as it does not involve thought and consideration of things (Luhmann 2000).

view of each agent, therefore creating the set $((\alpha_a, \beta_b, \gamma_c), \dots, (\alpha_a, \beta_b, \gamma_c), \dots, (\alpha_a, \beta_b, \gamma_c), \dots)$. In Marsh's formalism, Trust is separated into three distinct aspects, namely Basic Trust, General Trust and Situational Trust. In the following paragraph we elaborate on these concepts, and attempt to map them onto the BDT framework. Since we aim at measuring trust as a prosocial core factor of the human player, we adopt agent x for the remainder of our notations as the trusting end of the trust bond, i.e. the agent who is in need of determining whether to trust a given agent y or not. Therefore, from this point on, we will refer to the player simply as x. For the player x, knowing another character[11] y will be denoted as $K_x(y)$ and will take a true/false value (0, if x does not know y, and 1 otherwise). Therefore if $K_x(y) = 1$, we infer that x has already met y.

Basic Trust T_x^t refers to x's general trusting disposition (S. D. Boon & J. G. Holmes 1991) at a specific time t, and takes on values within the range $[-1, +1)$. The higher T_x^t is, the more trusting x is at time t. This metric is a general indicator of user disposition to trust a character that has only recently been encountered. This disposition is then further adjusted according to past experiences, i.e. dependent on what has happened to x in the past. Therefore, we will accept that good experiences lead to greater disposition to trust, and vice versa (S. D. Boon & J. G. Holmes 1991). As this trust aspect does not imply any trust directed towards any other agent or depending on a particular situation, we can map T_x^t on the BDT framework in all cases in which the Target's *Specificity* is general. Furthermore, the subject touches upon the Cognitive/Affective domain, demonstrating how x's beliefs stand that others demonstrate the three bases of trust. This can indicate a measure of optimism/pessimism, as optimistic characters will expect the best in all things and be always hopeful for the outcome of situations, while pessimists act in the exact opposite way (Marsh 1994b). A flow diagram showcasing this simplest case of trust is presented in Figure 39. Here, the final decisions represent a general disposition towards behaving in that particular manner. Also, distrust and no trust (or zero trust) are two different concepts (which are better explained in the next paragraph).

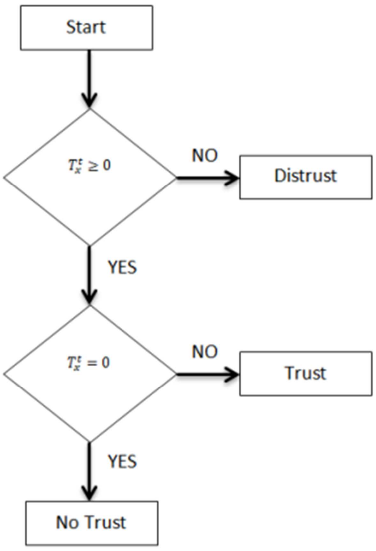


Figure 39: Flow diagram for agent x Basic trust T_x^t.

[11] The word character is used with an equivalent meaning of player.

General Trust refers to trust in other agents, therefore $T_x(y)^t$ is a measure of how much x trusts in y at a specific time t, and like basic trust, it will take up values in the range $[-1,+1)$. The values of general trust are a basic indication of the trust x has in another character y at any time. A value of 0 means x has no trust in y, which could be the outcome of past transactions appraised both positively and negatively or simply because x has not yet met y , i.e. $K_x(y) = 0$. On the other hand, a value of -1 reveals a strong disposition of x to generally distrust y, which is indicative that they have met before i.e. $K_x(y) = 1$, and x has drawn this conclusion through any number of negative experiences x has come into in the past by making a decision to trust in y. A flow diagram for general trust can be seen in Figure 40.

We can already map these values onto BDT by setting the Target's *Specificity* to a specific agent y. Furthermore, the general trust value represents the probability that x will behave as if he trusts y, i.e. that x will expect that y will behave according to x's best interest and will not attempt to harm x. This is a direct reference to the Cognitive/Affective and Behavior-dependent Domains of the BDT framework, as well as the Behavior-enacting domain when viewed from y's perspective (i.e. x's own estimate of how much it is trusted by y), as defined in the previous paragraph. Since General trust can be defined across all Domains of the BDT framework, we will adopt it as a measurement for each character in the game with whom x interacts with. It is the single, general measure of trust x has in y, and is built across time, from the moment the two agents meet up until time t.

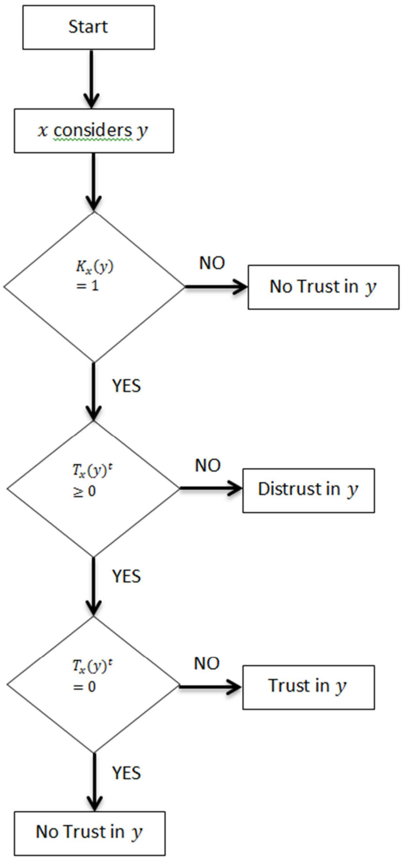


Figure 40: Flow diagram for agent x General trust $T_x(y)^t$ in agent y.

Situational Trust refers to the amount of trust x has in y in situation α. Similar to basic and general trust, situational trust $T_x(y, a_x)^t$ is defined within the $[-1, +1)$ space. This will be the most important aspect of trust in cooperative situations, as it provides a measure of trust in another to engage in cooperation to resolve a situation. If this measure is above a specific threshold, then cooperation ensues. A flow diagram is shown in Figure 41.

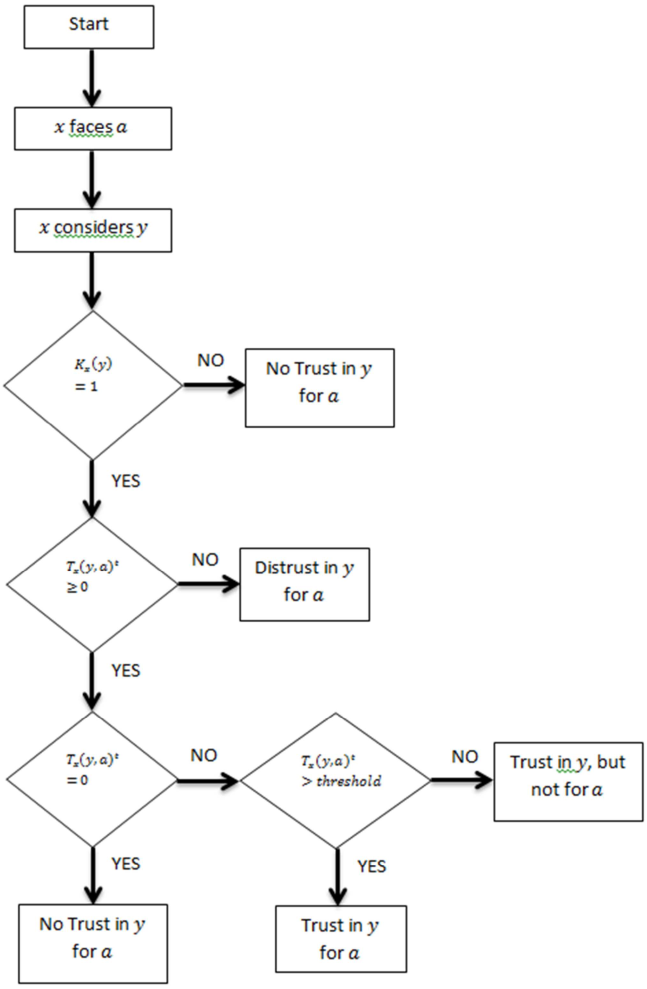


Figure 41: Flow diagram for agent x Situational trust $\mathrm{T_x(y,a)^t}$ in agent y for a situation a.

As is the case with the general trust aspect, situational trust in the scope of the project refers to trust in each one of the three bases of trust, as defined in the BDT framework. Therefore, Eq. (3) holds for each base, namely *Honesty* $(T_x^H(y, \eta)^t, \eta \in \boldsymbol{H})$, *Reliability* $(T_x^R(y, \rho)^t, \rho \in \boldsymbol{R})$ and *Emotional Trust* $(T_x^E(y, \varepsilon)^t, \varepsilon \in \boldsymbol{E})$.

Having determined the situational trust $T_x(y, a_x)^t$ in y, x can consider whether or not to cooperate, or otherwise engage in trusting behavior. To determine the answer, threshold values for trust will be defined. (Marsh 1994b) defines the cooperation threshold as a barrier for overcoming trust issues. If the situational trust is above a cooperation threshold $W_x(a)^t$ cooperation will occur, otherwise cooperation will not occur. When x's trust in y with regard to situation a is less than the cooperation threshold (denoted as $W_x(a)$), but still larger than 0, we define the concept of *untrust*. On the other hand, an active judgement in the negative intentions of another formalizes the concept of *distrust*, or that x believes that y will act contrary to their best interests in the situation a. A graphical representation of the continuous values of trust and where definitions are placed is shown Figure 42.

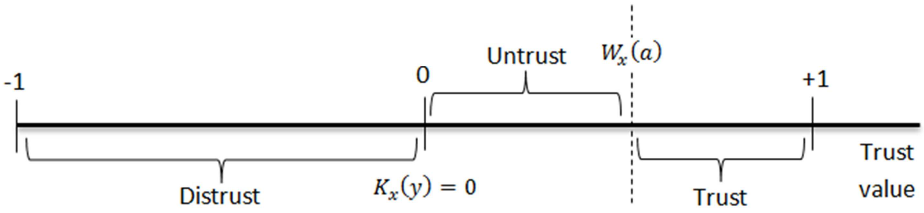


Figure 42: Trust continuum, from Distrust to Trust and Untrust.

We have laid out a strong foundation on which we aim to build our computational model, which will be presented in future editions of this deliverable (D3.3, D3.4). It is important however to note that we foresee using this computational model to compute x's general trust aspects for each base and each agent/person y in the game, as a post-hoc metric to the actual decision to cooperate with y based on trust. This will provide us with ground truth data for that particular player's relationships with other agents in the game. However, x's dilemma at any given time t, where it has to decide whether to trust in y during a situation a, will only be measured as an estimate. Therefore, while processing a new situation unfolding at current time t, x has to be rated according to the trust it demonstrates across all Domains. This means, that we can compute what x's trust value in this particular case should be, then monitor x's decision on whether to trust in y or not, re-evaluate its general trust aspects to get new ground truth values for future cases at time $t+1$ and propose adaptations according to whether x acted out on its expected trust factor at time t. More on this adaptation will be presented in D4.1.

- **Correlating to the SAVE Framework**

Assuming the giver is the person who acts in a prosocial fashion, we can expand the formula of the SAVE framework (Eq. 1), and moving the total perceptions of cost into one part of the inequality, we end up with the following representation of the SAVE framework:

$$M \times \left(D \times \left(1 + B_{self}\right) + K \times B_{recipient} - C_{innaction}\right) > C_{action} \Leftrightarrow$$

$$M \times \left(D + \left(\mathrm{D} \times B_{self}\right) + K \times B_{recipient} - C_{innaction}\right) > C_{action} \Leftrightarrow$$

$$(M \times \mathrm{D}) + \left(M \times \mathrm{D} \times B_{self}\right) + (\mathrm{M} \times K \times B_{recipient}) - \left(\mathrm{M} \times C_{innaction}\right) > C_{action} \Leftrightarrow$$

$$\left((M \times D) \times \left(1 + B_{self}\right)\right) + (\mathrm{M} \times K \times B_{recipient}) - (\mathrm{M} \times C_{innaction}) > C_{action} \Leftrightarrow$$

$$\left((M \times D) \times \left(1 + B_{self}\right)\right) + \left(M \times K \times B_{recipient}\right) > \left(C_{action} + (M \times C_{innaction})\right) \quad (2)$$

According to the definitions of M, D, and B_{self}, the first left-hand term $\left((M \times D) \times \left(1 + B_{self}\right)\right)$ can be summarized as the giver's psychological stance against its own perception of the situation, i.e. it is a self-assessment of the utility to be gained from a situation, weighted by the importance (a subjective measure which corresponds to the definition of D) and the general influence the agent perceives from its society in behaving prosocially, i.e. "is it the right thing to do?". Accordingly, considering the definitions of K and $B_{recipient}$, the second term on the left-hand side of the formula

$\left(M \times K \times B_{recipient}\right)$ corresponds to the giver's perception of the recipients capability and stance against that particular individual, weighted by that individual's utility of the situation. The right-hand term $\left(C_{action} + \left(M \times C_{innaction}\right)\right)$ corresponds to the perceived cost of acting and not acting, which may be geared towards a consideration of the risks involved in each situation. Since we agree with the scientific literature that the act of trust is in fact prosocial behavior, according to (Keltner et al. 2014), trust should be governed by Eq. (2). If the condition holds true, prosocial behaviour, i.e. trust, will ensue. Since we refer to the giver as agent x and the recipient as agent y for the majority of this text we will re-write Eq.(2) in an appropriate fashion:

$$\left((M_x \times D_x) \times (1 + B_x)\right) + \left(M_x \times K_x \times B_y\right) > \left(C_x^a + \left(M_x \times C_x^{\neg a}\right)\right) \rightarrow f(x, y, a) \quad (3)$$

Where the subscript x denotes that the given variable is considered from x's point of view, and the superscript a is used to indicate that situation a unfolds. Similarly, $\neg a$ refers to the situation a not unfolding, since x will not in fact, cooperate with y. This is denoted as $f(x, y, a)$, i.e. x trusts in y for situation a. Eq. (14) basically boils down to:

$$Perception_x^a + Perception_x^y > Perception_x^{risk} \rightarrow f(x, y, a) \quad (4)$$

i.e. if my perception of the situation a and my perception of the candidate trustee y is above a threshold set for my perception of the risks involved for situation a happening or not happening, I can trust in y.

This representation of the SAVE framework is very much likely in tune with our computational model, since both state that x has to overcome some uncertainty factor (cooperation threshold) in order to decide to trust in y. We are encouraged by this revelation, and will further explore the correlation between our emerging computational model and the SAVE framework in future editions of this deliverable.

Cooperation

Cooperation has been discussed under different fields and with different angles. “game theory” and “cooperative game theory” focuses on the strategy that agents put in play to maximize their payoff. Other approaches focus on the observation of cooperation to identify reasons behind the cooperation and the factors influencing it. Cooperation models are also quite dependant on the context in which they are used.

The reason for cooperating is at a first glance always derived from a cost benefit analysis. The benefit may or may not be for the helper. According to (Lehmann & Keller 2006) the difference between altruism and cooperativeness is in the direct benefit for the helper. If the benefit is positive it is defined as cooperation, if it is negative it is defined as altruism. The authors identify four general situations where helping is favoured of which two are related to cooperation. “*The first is when the act of helping provides direct benefits to the FI*[12] *that outweighs the cost of helping (i.e. there are direct benefits). […] . The second situation is when the FI can alter the behaviour response of its partners by helping and thereby receives in return benefits that outweigh the cost of helping. In both situations, the helping act is cooperative as it results in an increase of the fitness of both the FI and its partners. A difference, however, is that in the first situation the increase of the FI's fitness is because*

[12] In the paper “FI” stands for focal individual. In the contest of the model used in PsL is simply the player in charge of taking the decision.

of its own behaviour while in the second situation it results from the behavioural change induced in its partner(s).”

In particular the considered scenario is defined as

- “non-cooperative”[13,14] according to (Peleg & Sudholter 2005) because there is not agreed binding among players before the game starts.
- “non zero sum”[15]. In game theory and economic theory, a zero-sum game is a mathematical representation of a situation in which each participant's gain (or loss) of utility is exactly balanced by the losses (or gains) of the utility of the other participant(s). If the total gains of the participants are added up and the total losses are subtracted, they will sum to zero. In our scenario better explained below, this condition is not verified.

Our model considers cooperation scenarios as defined above, and in particular considers private and public good scenario as following:

- A succession of decision points exist that leads to an “all win” or “all lose” scenario.
- In each decision point a user
  - can decide to help another user sacrificing part of his private resources and so maintaining unaltered the probability of the “all lose” scenario
  - can decide not to help another user preserving his private resources and so increasing the probability of the “all lose” scenario
- The all lose scenario happens after a fixed amount of decision point has been resolved with a “not to help” choice

A common way for defining the contribution to a common good is to calculate the marginal contribution of a user. In this case we can identify the contribution as the payment for the maintaining of the “all lose” probability.

Let be

- TMAX the maximum number “not to help” decisions before the “all lose” scenario happens.
- T the number of already happened “not to help”.
- DEC the total number of decision points. This parameter is calculated as average of values in already executed scenarios.

At a given decision point, the probability that “all lose” scenario is verified is

- P”All Lose”= $1 - \frac{\mathrm{T_{MAX}} - \mathrm{T}}{\mathrm{DEC}}$

with TMAX≥T and T=TMAX defining the condition of “all lose”.

When TMAX-T=1 deciding to help or not to help is a pure cognitive exercise, because not helping means to lose everything also for the helper (“all lose” scenario).

[13] Note that ‘cooperative’ and ‘non-cooperative’ are technical terms and are not an assessment of the degree of cooperation among agents in the model: a cooperative game can as much model extreme competition as a non- cooperative game can model cooperation.

[14] https://en.wikipedia.org/wiki/Non-cooperative_game

[15] https://en.wikipedia.org/wiki/Zero-sum_game

When TMAX-T=2 a player may decide to preserve his private goods relying on the fact that someone else will be forced to help the next decision point to avoid the “all lose” condition (see point above). This situation is similar to the well-known free-rider problem in game theory.

The scenario discussed above is characterized by a given situation when it is obvious that cooperate is the only choice and a concept of “distance” from this situation. The distance is the number of subsequent decision points in which it is affordable to have the “all lose” scenario[16].

As for the cooperation measurement the idea is therefore to reward (i.e. consider more cooperative) the action of helping the more it is far from the “pure cognitive” exercise.

The function expressing the positive cooperativeness of a user is the counting of times he decides to help weighted by inverted probability of the “all lose” scenario $\frac{\mathrm{T_{MAX}-T}}{\mathrm{DEC}}$

Positive Cooperative Level = $\sum_{i}^{n} \frac{\frac{\mathrm{Tmax-T_i}}{\mathrm{DEC}}}{\mathrm{n}}$ With n= number of “help”

Analogously a Negative Cooperative Level for a user represents the free riding scenario and it counts the time the user decides not to help weighting them with the distance to the “all lose” scenario at the time of the choice $\frac{\mathrm{T_{MAX}-T}}{\mathrm{DEC}}$.

Negative Cooperative Level = $- \sum_{i}^{n} \frac{\frac{\mathrm{Tmax-T_i}}{\mathrm{DEC}}}{\mathrm{n}}$ With n= number of “not help”

The Cooperative level is therefore defined as the algebraic sum of Positive and Cooperative level. This value may be normalised in the model so that it becomes independent from the “*DEC*” parameter.

This is a simplistic approach to extract values of cooperation from that will be tested and assessed, but it is considered a fair representation of a cooperation value as derived from in game data. The model will be completed adding to it a representation of emotions.

The proposed model follows the approach of the SAVE model where a cost/benefit balance is applied. In this simplified version there is the concept of personal cost, personal benefit and global benefit that are combined and evaluated after any decision taken by the player. A direct mapping of the terms of this model against the SAVE model is more difficult because the SAVE model describes a propensity to be prosocial while the goal of this operational model is to provide an actual measurement of cooperation and the constant in the SAVE model are not yet defined.

Fairness

This model is directly inspired from the SAVE framework presented at the beginning of this section.

$$\mathrm{M} \times (\mathrm{D} \times (1 + \mathrm{B_{self}}) + \mathrm{K} \times \mathrm{B_{recipient}} - \mathrm{C_{innaction}}) > \mathrm{C_{action}}$$

[16] Different strategies can exist in this type of game. A possible one (to be verified by the actual experiment) is that each player will decide to free-ride (preserving his own private goods) up until the last possible moment (i.e. the obvious decision point). Deviation from this strategy will give us information on how the human player acts in reality.

In the following we will discuss the values of the constants and how we may calculate it in the case of a fairness based game.

As said **M** corresponds to the social momentum for acting prosocially, or the influence of the socio-cultural milieu. Values ranging from 0 to 1 show social resistance. Values from 1 to infinity show a positive influence of the milieu. A value of 1 corresponds to an absence of social influence. We do not know yet what the value will be for this as the literature shows no clear relationship between Socio Economic Status (SES) or culture and children's prosocial behaviour (see report D2.1). We might however get this value from the pilot studies we will conduct. Waiting to have a clear value of the influence of the SES in our games, we recommend leaving this value at 1 for no influence. In the pilot studies, we should make sure we record the social status of the area where we conduct the testing (if not of the children).

D represents individual differences in prosociality and situational factors. We will be able to get this value from (a) the data we collect on individual differences (e.g. personality and attachment style) and (b) the situational factors such as the instruction of the game (who will know about the student being prosocial etc; these instructions will be different for each games). There is no a priori value for this.

In order to simplify the influence of individual differences and situational factors, we actually recommend separating D into I and S such as **D= (I+S).** This way, we can clearly explain the influence of **I**, the personality and attachment style that we will get from the questionnaires; and **S**, which will depend on the instructions.

I itself could be split in **I= (A +P), A** standing for attachment style and **P** for personality. These are the two traits we will be measured in this project. However, as a note, **I** could actually be a combination of more than just these two traits.

Because research has shown that securely attached children are more socially competent and have better friendship quality (see D2.1 report, p32-33), **A** could for instance take a value of 1 to infinity for securely attached children and negative values for non-securely attached children. We could reduce infinity to 100 by having a scale going from -100 to +100 on how secured these children are (from the questionnaires). However, note that the ‘0’ value should be omitted. So in fact we should go from -100 to -1 for insecurely attached children and +1.1 to +100 for securely attached children.

The same could be done for agreeableness (calculated on the personality questionnaire) for the value **P**.

The value of **S** will largely depend on the instructions of each game, but below is a suggestion for likely scenarios. Prosocial behaviour is highly contagious (see (Nowak & Roch 2007; Christakis & Fowler 2009)). That is, if you are in an environment where everyone shares and is fair to each other, you will likely be more prosocial yourself. Backstories can help us determine this function. If the children know that other children will know the outcome of their decisions and if all the other children are prosocial, then D should have a positive value (from 1.1 to 100 for instance). In the opposite case, negative values should be used if the context describes people who are not being prosocial towards each other.

Bself corresponds to the perceived benefit to the self. These benefits can be indefinite and can take many forms. It can for instance be: money, friends, reward, making a difference, being equal, etc.The instruction and context of the games should help us to determine what the benefits might be (if anonymous setting, making friends will not be a benefit; but making a difference could be etc).

Brecipient is the benefit another person can receive from the prosocial action (e.g money, friends etc). This is how much money the receiver makes in the giver-receiver game for instance.

Specifically, we can summarise the action of **Bself** and **Brecipient** by the Figure 43 below. In this figure, the benefit for the self and the recipient take the form of: concern for substance (sweets in this case: what is the maximum amount of sweets I can get?) and concern for relationship (what is the best amount that I can give to keep my relationship with my friends). The expectation however will be different if the game is played anonymously (no concern for relationship) or if the game is played in collaboration or not. Indeed, if the game is played in collaboration, the monetary value (the sweets) that is considered 'fair' will likely be higher than if the game is play independently (why would they give half of their gain if the other player has not helped to collect the sweets?). The results from the pilot study will help us determine what the values are for this specific game in these specific situations.

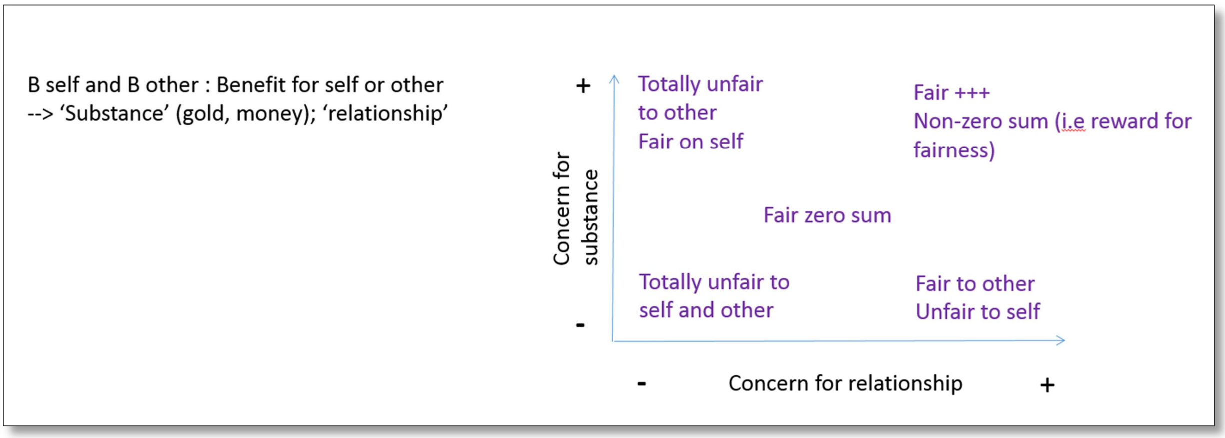

B self and B other : Benefit for self or other
--> 'Substance' (gold, money); 'relationship'
+
Concern for substance
-
Totally unfair to other Fair on self
Fair +++ Non-zero sum (i.e reward for fairness)
Fair zero sum
Totally unfair to self and other
Fair to other Unfair to self
-
Concern for relationship
+

Figure 43: Bself and Bother in the Giver-Receiver game.

K is the giver's biases and perceptions of the specific recipient, which range from positively valenced preferences (e.g., in-group members) to negative values that reflect adversarial stances toward others (e.g., competition, intergroup biases). The instruction context will help us determine this again: Do they have to be generous to their friends or enemies? Is the other player a close friends or just a classmate? In the case of the anonymous giver-receiver game, this should be 1 so it has no influence. See Figure 2 for values of K for a game where the other player interacting is an enemy or a friend.

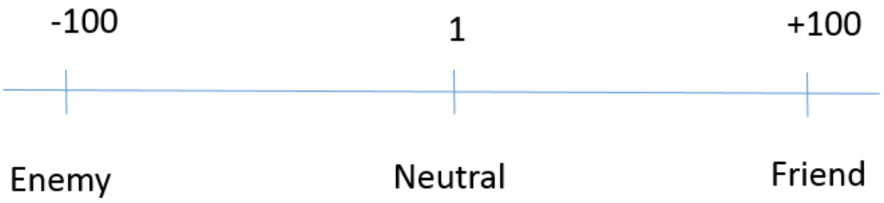


Figure 44 K, the giver's bias and perception

Cinaction corresponds to the cost or perceived consequences of not acting. This can take the form of guilt for the individual, or reputation loss, gossip etc at the group level. Again, the context and instruction will help determine what will have an influence. This value could for instance be 0 in the anonymous giver-receiver game for the group level. But might not be 0 is the giver feels guilty for not acting for instance. The inclusion of emotions in our model will help determine whether **Cinaction** can take the form of guilt, which will help get a better estimate of their prosocial skills.

Caction is the cost to the self for acting prosocially. For example, prosocial behavior can involve the giving up of a valued resource (e.g., money) to benefit another. This is for instance how much sweets they give away in the giver-receiver game. They might also gain some positive emotions, feeling good about being generous and the fusion of emotions will help us have a better approximation of **Caction**. The value can for instance vary from 0 to 100 according to how many sweets they share. **Caction** is not only how much they share but how they feel about it etc. This is developed below.

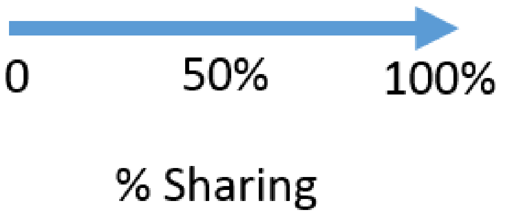


Figure 45 Caction: monetary part only

To determine the values for **Caction** and **Cinaction**, the pilot study will be of considerable help. To do so, it is important to add a few questions to the giver-receiver game to better capture the whole range of values that we are interested in.

Amendment 1:

After we ask the children to share the sweets or after they see how much the other player decided to give them, we should ask a few more questions considering:

- The emotional state of self
- The emotional state of the other
- What they think is fair, as a giver and a receiver.
- How they and the other person feels

For the receiver in particular, we should ask questions such as:

- What do you think would be fair for you to receive?
- How do you feel about the other person?
- Would you have shared more?
- Do you think they are being fair?
- How are you feeling right now?
- If they were your friend, should they have shared more or less?
- If they were your enemy, should they have shared more or less?

In return, we should ask the giver:

- How do you think the other person feels?
- How fair do you think you were on the other person?
- The other person thought you were fair/unfair and feel happy/sad. How do you feel about it?

- Do you think you should have shared more or less?
- Would you have shared more if they were your friend? (how much would you shared if they were your friend?)
- Would you have shared less if they were your enemy?

Another alternative would be to create a fictional character with a backstory making them being liked/dislike by the player. We could have two conditions, one with a friend, one with an enemy. This would create 8 conditions in total:

1. Play on your own against friend → Give
2. Play on your own against friend → Receive
3. Play on your own against enemy→ Give
4. Play on your own against enemy→ Receive
5. Play in collaboration against friend → Give
6. Play in collaboration against friend → Receive
7. Play in collaboration against enemy → Give
8. Play in collaboration against enemy → Receive

This might be slightly too long for the purpose of the testing in school. We could also maybe do groups of children. Some do Game 1 to 4, some 5 to 8 etc.

To summarise, it is currently difficult to have an a priori value for what the values of all these variables should be. However, we are hoping that the pilot study will help us get a better estimate for specific games. These estimates will have to be re-adjusted for new games with different instructions.

Appendix D: User graphical interface

ProsocialLearn designs and offers graphical interfaces to the different classes of the end-users. The interfaces will present a unified look and feel and they can be logically grouped in two sets: management interfaces and querying interfaces.

The management interfaces group all the functionality needed to i) deploy, publish, access games ii) create users and roles iii) manage logins and access rights.

The querying interfaces are described in the Description of Activities[17] as Student Learning and Assessment Dashboard (SLA Dashboard) and they group the functionalities that through appropriate queries on the available data present all the information to the end-users (e.g. the assessment of a given properties for a student across a set of games).

This chapter introduces the Student Learning and Assessment Dashboard, outlining its place within the ProsocialLearn project and system, and presents some initial design ideas for the first prototype. It focuses on a specific subset of end-users including:

- Teacher
- Student
- Parent
- Psychologist

It is also likely that an overall Admin user will be required, for general configuration and maintenance of the dashboard, setting up initial user accounts, etc.

Our initial focus is on use cases most relevant to the Teacher (as outlined in the previous section) including game lesson planning and presentation of game history. These will help to drive the design of the initial prototype for the SLA Dashboard, and will be further developed and added to as the project progresses.

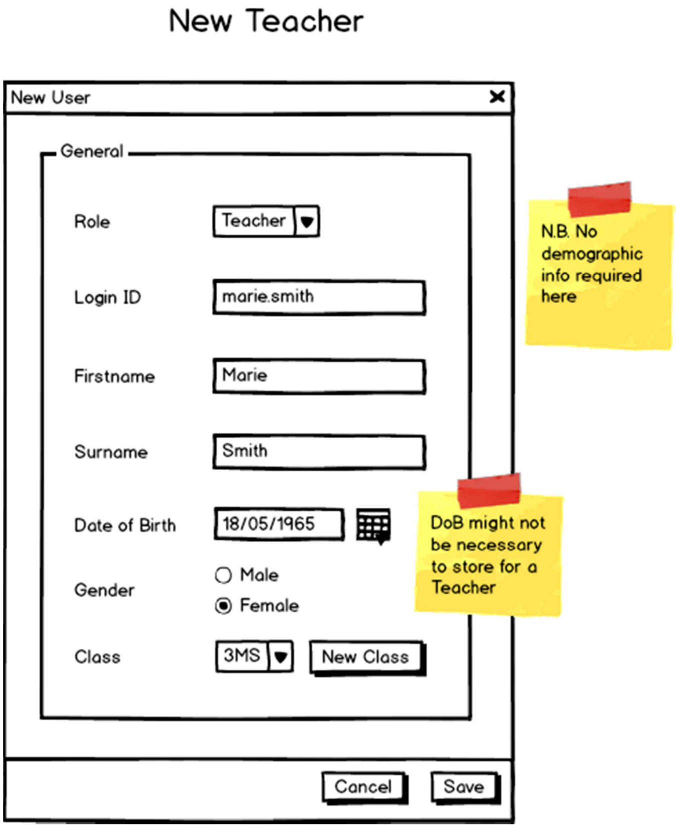
Creating a new Teacher user

Before a teacher can use the SLA Dashboard, they will require a login account. This may require some assistance from an Administrator in the first instance, or that a responsible teacher is also provided with an Admin role in the system.

Figure 46 shows a mock-up of the New User dialog, which will pop-up after selecting “New User” from the main dashboard menu (for example). Several different types of user can be created via this dialog, by selecting the appropriate Role (e.g. Teacher, Student, Parent, and Psychologist). In this case, we select “Teacher” as the role. The dialog will be context sensitive, and will therefore only show fields that are relevant to the particular type of user (for example only Student users will be shown input fields related to demographics, as we will see later).

The Teacher will be requested to enter general details such as Login ID, Firstname, Surname, Date of Birth and Gender. They will also be able to select the Class that they teach, or enter a new one via the “New Class” button. This feature is shown in the next section.

[17] ProsocialLearn Description of Activities (previous known as DoW)



Role: Teacher ▾

Login ID: marie.smith

Firstname: Marie

Surname: Smith

Date of Birth: 18/05/1965

Gender: ○ Male ● Female

Class: 3MS ▾

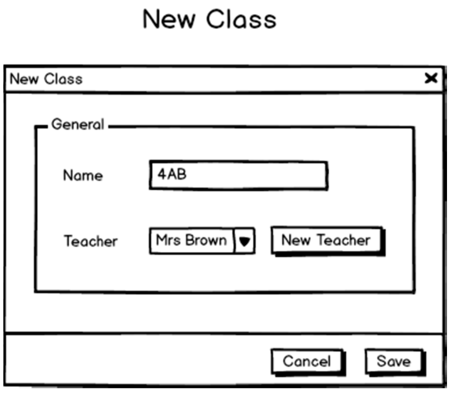
N.B. No demographic info required here

DoB might not be necessary to store for a Teacher

Figure 46: Creating a new Teacher

Creating a new Class

Figure 47 shows a dialog for creating a new Class, which simply requires a Name and a Teacher. Once classes are set up, Students can be created and added to their correct Class (as we will see in the next section).

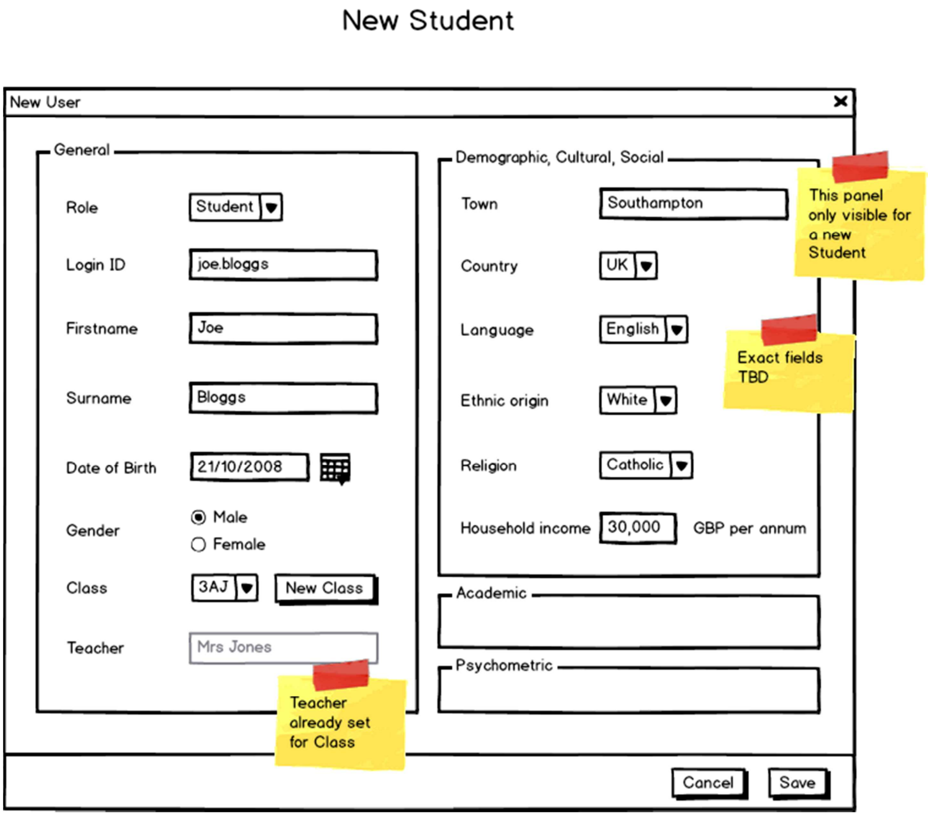


Name: 4AB

Teacher: Mrs Brown ▾

Figure 47: Creating a new Class

Creating a new Student user



New Student

New User ✕

General

Role: Student ▾

Login ID: joe.bloggs

Firstname: Joe

Surname: Bloggs

Date of Birth: 21/10/2008

Gender: ● Male ○ Female

Class: 3AJ ▾ New Class

Teacher: Mrs Jones

Demographic, Cultural, Social

Town: Southampton

Country: UK ▾

Language: English ▾

Ethnic origin: White ▾

Religion: Catholic ▾

Household income: 30,000 GBP per annum

Academic

Psychometric

Cancel Save

This panel only visible for a new Student

Exact fields TBD

Teacher already set for Class

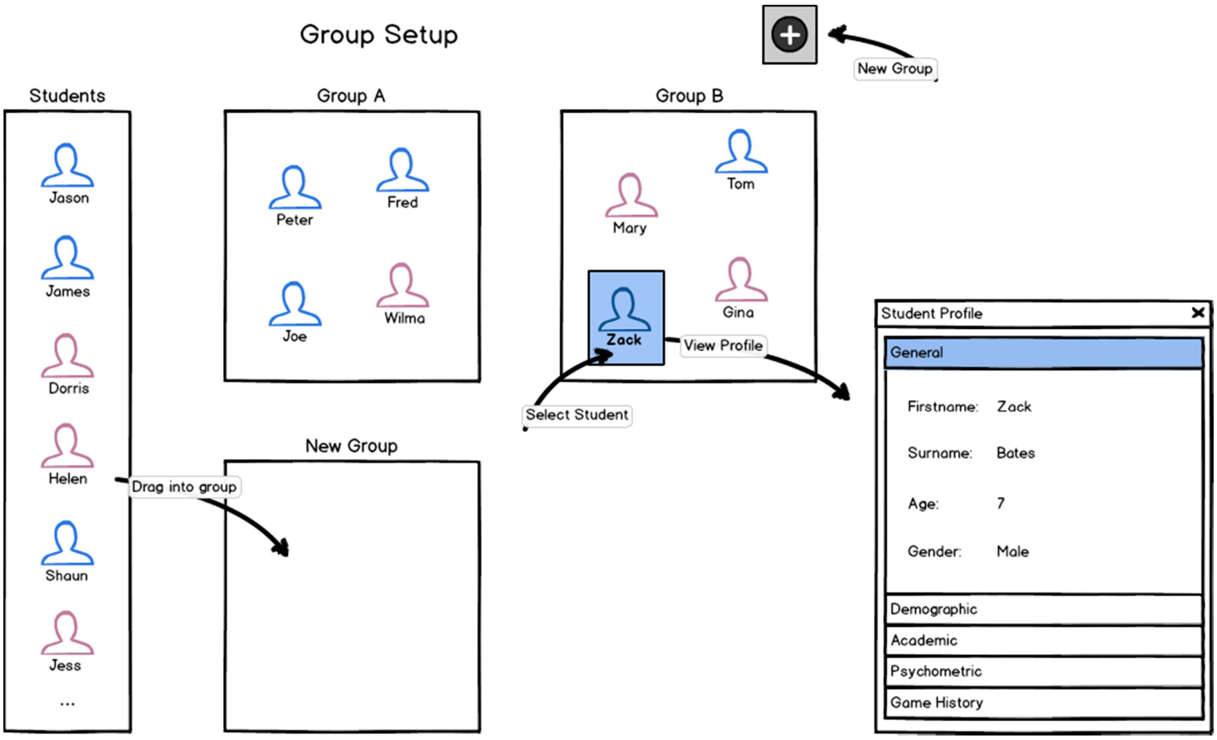
Figure 48: Creating a new Student

Figure 48 shows the dialog for creating a new Student user. This is similar to creating a Teacher, however additional input fields will be displayed for entering demographic, cultural, social data, etc. The exact fields required here will be clarified during the project.

Note that, when selecting a Student's Class, the Teacher field will be automatically populated, as this has already been assigned for the class.

Setting up Class Groups

In order to play prosocial games, a teacher will need to be able to organise the students in his/her class into various Groups (also known as cohorts). For example a particular game might require four participants, therefore the students will need to be put into groups of four. Figure 49 shows a mock-up of a proposed dashboard page for setting up Groups. Here, we start with a pool of Students (associated with the Teacher's Class). Using the GUI, the teacher will be able to create a new Group, then drag students into these. Students may be colour-coded according to gender, so the teacher can allocate students into same-sex or mixed groups, as required. By selecting individual Students, a teacher will be able to view their profiles, in order to facilitate their allocation to groups. Profiles may be viewed side by side. Once allocated to Groups, teachers may then schedule groups for playing games.



Firstname: Zack

Surname: Bates

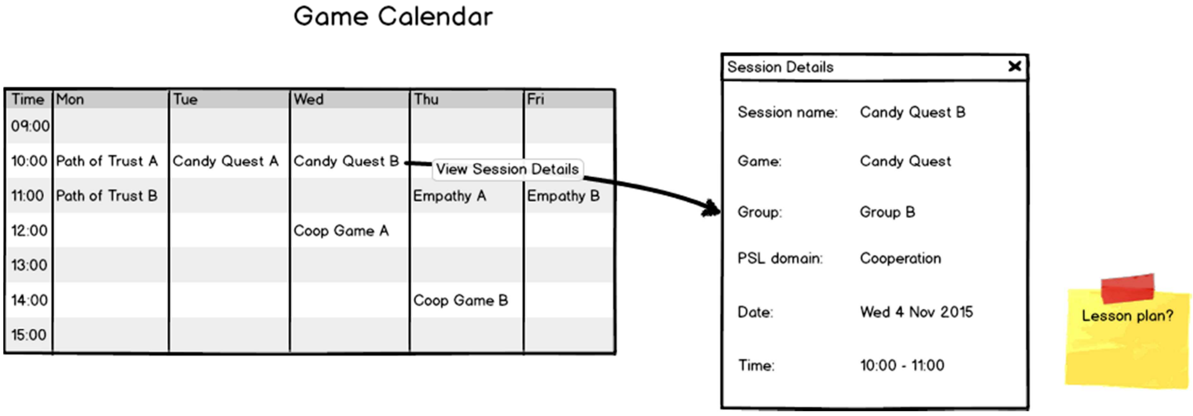
Age: 7

Gender: Male

Figure 49: Setting up a Group

Scheduling Game Sessions

It is likely that a teacher will need to schedule groups of students to play games within various sessions. For example, a school might have limited computing resources, so not all students in a class might be able to play games at the same time. Game sessions will also help teachers to organise their students to play a series of games, and subsequently to track their progress. Each session will have a name, along with an associated game and student group that will play in that session. A scheduled date and time will be provided. Ideally, game sessions will be displayed in some form of calendar for the teacher, as shown in Figure 50.



Time	Mon	Tue	Wed	Thu	Fri
09:00					
10:00	Path of Trust A	Candy Quest A	Candy Quest B		
11:00	Path of Trust B			Empathy A	Empathy B
12:00			Coop Game A		
13:00					
14:00				Coop Game B	
15:00					

Session name: Candy Quest B

Game: Candy Quest

Group: Group B

PSL domain: Cooperation

Date: Wed 4 Nov 2015

Time: 10:00 - 11:00

Figure 50: Setting up a Game Schedule

The Game Calendar will enable a teacher to quickly see when game sessions are scheduled. New sessions may be created directly in the calendar. In Figure 50, we have assumed that one Group plays a particular Game within a Game Session. For example, session "Candy Quest A" is set up for Group A, whereas "Candy Quest B" is for Group B. Teachers may wish to have other grouping options, e.g. having several groups of students playing simultaneously, during a particular session (i.e. time slot).

Another benefit of defining sessions in this way is that the teacher will maintain control over which students may play games at any given time. For example, a teacher would not want students to start playing a random game at some unspecified time. There is no point in an individual student playing a cooperation game on their own! A formal session would provide some form of access control, whereby a student could only play a particular game within a fixed time period.

A session would also have some form of state, e.g. "Scheduled", "Running" or "Complete"; it could automatically enter the "Running" state, once its time slot starts, then change to "Complete" once the time slot expires. Alternatively (or in addition), the teacher might have a direct control in the dashboard to start a session with a group of students (and then to stop it once the game has been played). In this case, it may be useful to store the actual start and end time of the game playing session, in addition to the original scheduled times. The exact mechanisms of game scheduling, students logging in and joining sessions, playing games, etc., will be further refined in the coming months.

Student Game History

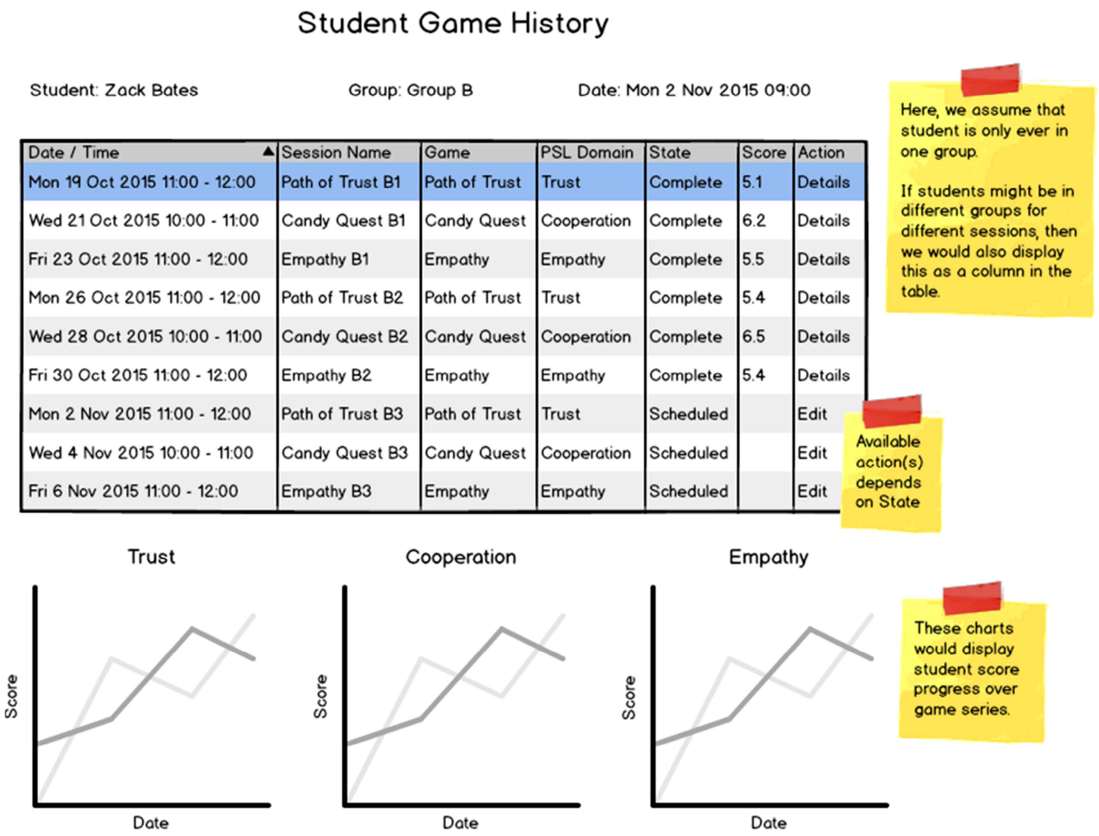
Once game sessions have been scheduled and played, the teacher will be in a position to review these as a summary or history. This will provide the teacher with a quick overview of sessions related either to a student or a group of students, showing:

- What game sessions are scheduled, running or completed
- What games have been played
- What prosocial domains have been evaluated through these games
- What prosocial domain "scores" that the student has achieved in each session

These features may be displayed in a table, as in Figure 51, which lists all sessions involving a particular student. It shows the student's name, group and the current date of this summary. The teacher would be able to sort the table by various columns, e.g. date, game played, etc. By selecting a particular session, the teacher could view further details of a completed game session (e.g. the low-level game data), or edit the schedule details (for a schedule that had not yet been run). Other options may also be available here.

Below the session summary table, it would be useful to display some charts to summarise aspects of the student's progress over a series of games. Here we propose to show charts for each PSL domain (e.g. Trust, Cooperation, and Empathy), i.e. how the scores progress over time. Other visual representations will be explored.

Along with a summary of an individual student's sessions and progress, it would be useful to be able to see figures for all students in a particular group. This is shown in Figure 52. Here, student scores are itemised in the table, and would also be displayed as distinct lines on the charts below.



Student Game History

Student: Zack Bates

Group: Group B

Date: Mon 2 Nov 2015 09:00

Date / Time ▲	Session Name	Game	PSL Domain	State	Score	Action
Mon 19 Oct 2015 11:00 - 12:00	Path of Trust B1	Path of Trust	Trust	Complete	5.1	Details
Wed 21 Oct 2015 10:00 - 11:00	Candy Quest B1	Candy Quest	Cooperation	Complete	6.2	Details
Fri 23 Oct 2015 11:00 - 12:00	Empathy B1	Empathy	Empathy	Complete	5.5	Details
Mon 26 Oct 2015 11:00 - 12:00	Path of Trust B2	Path of Trust	Trust	Complete	5.4	Details
Wed 28 Oct 2015 10:00 - 11:00	Candy Quest B2	Candy Quest	Cooperation	Complete	6.5	Details
Fri 30 Oct 2015 11:00 - 12:00	Empathy B2	Empathy	Empathy	Complete	5.4	Details
Mon 2 Nov 2015 11:00 - 12:00	Path of Trust B3	Path of Trust	Trust	Scheduled		Edit
Wed 4 Nov 2015 10:00 - 11:00	Candy Quest B3	Candy Quest	Cooperation	Scheduled		Edit
Fri 6 Nov 2015 11:00 - 12:00	Empathy B3	Empathy	Empathy	Scheduled		Edit

Here, we assume that student is only ever in one group.

If students might be in different groups for different sessions, then we would also display this as a column in the table.

Available action(s) depends on State

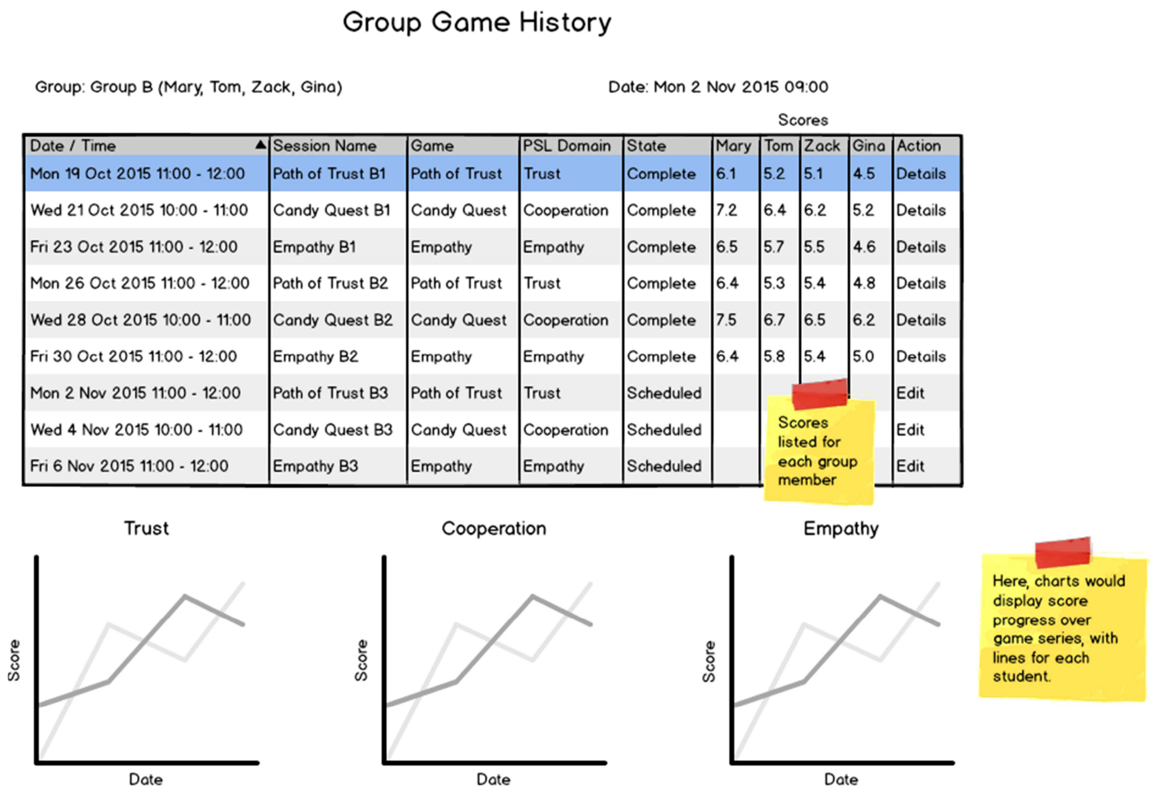
Trust

Cooperation

Empathy

These charts would display student score progress over game series.

Figure 51: Displaying a Student Game History



Group Game History

Group: Group B (Mary, Tom, Zack, Gina)

Date: Mon 2 Nov 2015 09:00

					Scores				
Date / Time ▲	Session Name	Game	PSL Domain	State	Mary	Tom	Zack	Gina	Action
Mon 19 Oct 2015 11:00 - 12:00	Path of Trust B1	Path of Trust	Trust	Complete	6.1	5.2	5.1	4.5	Details
Wed 21 Oct 2015 10:00 - 11:00	Candy Quest B1	Candy Quest	Cooperation	Complete	7.2	6.4	6.2	5.2	Details
Fri 23 Oct 2015 11:00 - 12:00	Empathy B1	Empathy	Empathy	Complete	6.5	5.7	5.5	4.6	Details
Mon 26 Oct 2015 11:00 - 12:00	Path of Trust B2	Path of Trust	Trust	Complete	6.4	5.3	5.4	4.8	Details
Wed 28 Oct 2015 10:00 - 11:00	Candy Quest B2	Candy Quest	Cooperation	Complete	7.5	6.7	6.5	6.2	Details
Fri 30 Oct 2015 11:00 - 12:00	Empathy B2	Empathy	Empathy	Complete	6.4	5.8	5.4	5.0	Details
Mon 2 Nov 2015 11:00 - 12:00	Path of Trust B3	Path of Trust	Trust	Scheduled					Edit
Wed 4 Nov 2015 10:00 - 11:00	Candy Quest B3	Candy Quest	Cooperation	Scheduled					Edit
Fri 6 Nov 2015 11:00 - 12:00	Empathy B3	Empathy	Empathy	Scheduled					Edit

Scores listed for each group member

Here, charts would display score progress over game series, with lines for each student.

Figure 52: Displaying a Group Game History