VLSS

Journal of Visual Languages and Sentient Systems

Volume 2, October, 2016

Journal of
Visual Languages and Sentient Systems

An Open Access Journal published by

KSI Research Inc.

156 Park Square Lane, Pittsburgh, PA 15238 USA

Proceedings preparation, editing and printing are sponsored by KSI Research Inc.

Journal of

Visual Languages and Sentient Systems

Volume 2, 2016

Table of Content

INFORMATION FOR AUTHORS

The Journal of Visual Languages and Sentient Systems (VLSS) is intended to be a forum for researchers, practitioners and developers to exchange ideas and research results, for the advancement of visual languages and sentient multimedia systems. Sentient systems are distributed systems capable of actively interacting with the environment by gathering, processing, interpreting, storing and retrieving multimedia information originated from sensors, robots, actuators, websites and other information sources. In order for sentient systems to function efficiently and effectively, visual languages may play an important role.

VLSS publishes research papers, state-of-the-art surveys, review articles, in the areas of visual languages, sentient multimedia systems, distributed multimedia systems, sensor networks, multimedia interfaces, visual communication, multi-media communications, cognitive aspects of sensor-based systems, and parallel/distributed/neural computing & representations for multimedia information processing. Papers are also welcome to report on actual use, experience, transferred technologies in sentient multimedia systems and applications. Timely research notes, viewpoint articles, book reviews and tool reviews, not to exceed three pages, can also be submitted to VLSS.

Manuscripts shall be submitted electronically to VLSS. Original papers only will be considered. Manuscripts should follow the double-column format and be submitted in the form of a pdf file. Page 1 should contain the article title, author(s), and affiliation(s); the name and complete mailing address of the person to whom correspondence should be sent, and a short abstract (100-150 words). Any footnotes to the title (indicated by *, +, etc.) should be placed at the bottom of page 1.

Manuscripts are accepted for review with the understanding that the same work has not been and will not be nor is presently submitted elsewhere, and that its submission for publication has been approved by all of the authors and by the institution where the work was carried out; further, that any person cited as a course of personal communications has approved such citation. Written authorization may be required at the Editor's discretion. Articles and any other material published in VLSS represent the opinions of the author(s) and should not be construed to reflect the opinions of the Editor(s) and the Publisher.

Paper submission should be through https://www.easychair.org/conferences/?conf=dmsvlss2017

For further information contact: vlss@ksiresearch.org

Recommender Systems and Social Networks: an application in Cultural Heritage

Flora Amanto, Vincenzo Moscato, Antonio Picariello, Giancarlo Sperlí
Dip. di Ingegneria Elettrica e Tecnologie dell'Informazione
University of Naples "Federico II", Naples, Italy
{flora.amato, vmoscato, picus, giancarlo.sperli}@unina.it

Abstract

In the last decade Recommender Systems *have become useful tools helping users to find "what they need" from considerable amount of data. One of the more obvious applications of such systems in the Cultural Heritage domain is to assist users when visiting cultural environments (such as museums, archaeological sites, old town centers and so on), providing a multimedia guide that is able to dynamically suggest relevant information available in multiple web repositories (e.g. multimedia sharing systems and on-line social networks). In this paper, we propose a novel recommendation approach that combines several aspects of users - i.e. their* preferences *(usually in the shape of items' metadata) and* interactions *within a social community modeled using hypergraphs - together with items'* multimedia features *and* context information *within a general framework that can support different applications (touristic guiding services for museums, visiting paths recommendation for old town centers and archeological sites, etc.). Preliminary experiments on user satisfaction show how our approach provides very promising and interesting results.*

1 Introduction

The development and promotion of worldwide Cultural Heritage using Information and Communication Technologies (ICT) represent nowadays an important research issue with a variety of potential applications.

In the last decade, such technologies have radically changed the purpose of Cultural Heritage exhibitions that is rapidly moving from an old vision, providing a tourist with static information consisting of a large amount of cultural signs, to novel *personalized services*, matching the visitors' personal goals and behaviors by considering their cultural needs and preferences and context information.

Indeed, users' experience could be surely enhanced if, instead of using classic "tourist" devices, they could be embedded in a cultural environment with a number of functionalities for representing the relevant information derived from the available digital sources, such as text descriptions, pictures, and videos. In this way, tourists would be given the opportunity of enjoying multimedia stories in real time, thus enriching their cultural knowledge.

From the other hand, we are assisting to an explosive and amazing increase of digital information, and as a consequence, more and more huge data collections of different nature are widely available and have constrained users necessarily to deal with this ocean of information to find "what they need". In particular, on-line social networks (e.g. Facebook) and multimedia sharing systems (e.g. YouTube, Flickr, Panoramio, Instagram, etc.), together with open digital libraries and archives (e.g. DBpedia), constitute the main multimedia information sources that can be considered "useful" for tourists when they visiting cultural environments such as museums, archaeological sites, old town centers and so on.

As well known, *Recommender Systems* have been introduced to facilitate the browsing of such collections, thus realizing the transition in the Web from the *search* to the *discovery* paradigm.

Generally, recommender systems help people in retrieving information that match their preferences by recommending products or services from a large number of candidates, and support people in making decisions in various contexts: what items to buy, which movie to watch, which music to listen, what travels to do, or even who they can invite to their social network, just to make some examples [24, 25].

One of the more obvious applications of such systems in the Cultural Heritage domain is to assist users

DOI reference number: 10.18293/DMS2016-001

when visiting cultural environments, providing a *multimedia guide* that is able to dynamically suggest relevant information available in multiple web repositories.

Formally, a recommender system deals with a set of *users* $U = \{u_1 \ldots, u_m\}$ and a set of *items* $O = \{o_1, \ldots, o_n\}$. For each pair (u_i, o_j), a recommender can compute a *score* (or a *rank*) $r_{i,j}$ that measures the expected interest of user u_i in item o_j (or the expected utility of item o_j for user u_i), using a *knowledge base* and a *ranking* algorithm that generally could consider different combinations of the following characteristics: (i) user preferences and past behavior, (i) preferences and behavior of the user community, (iii) items' features and how they can match user preferences, (iv) user feedbacks, (v) context information (i.e. user location, observed items, weather and environmental conditions, etc.) and how recommendations can change together with the context.

In the literature, surveys on recommender systems usually classify the different kinds of approaches in four main categories: *content-based* [22, 29, 30] (with their extensions to deal multimedia data and their features [19, 13, 20]) , *collaborative filtering* [2, 29, 23, 17] (with their customizations to take into account social elements as user reviews and opinions [32, 28, 18, 27, 11, 21, 9]), *hybrid* [26] and *context aware* [10, 15, 16] techniques. Finally, a recent category of recommenders, named *Large Scale Recommender Systems* (LSRS) [31], calls for new capabilities of such applications to deal with very large amount of data with respect to scalability and efficiency issues.

In our opinion, modern recommending applications have to take into account in some way all the above characteristics to provide useful and reliable recommendations both for virtual and physical environments. To this goal, the last generation of recommender systems is usually composed by one or more of the following components [25].

A *pre-filtering* module that selects for each user u_i a subset $O_i^c \subset O$ containing items that are good candidates to be recommended; such items usually match user preferences and needs.

A *ranking* module that assigns w.r.t. user u_i a rank $r_{i,j}$ to each candidate item o_j in O_i^c using the well-known recommendation techniques (i.e., *content-based*, *collaborative filtering* and *hybrid* approaches) that can exploit in several ways items' features and users' preferences, feedbacks (in the majority of cases in terms of *ratings*) and behavior.

A *post-filtering* module that dynamically excluds, for each user u_i, some items from the recommendations' list; in this way, a new set $O_i^f \subseteq O_i^c$ is obtained on the base of user feedbacks, other contextual information (such as data coming from the interactions between the user and the application) and possible additional constraints.

In this paper, we propose a novel recommendation approach that combines several aspects of users - i.e. their *preferences* (in the shape of items' metadata) and *interactions* (user to user and user to content) within a social community modeled using hypergraphs - together with items' *multimedia features* and *context information* within a general framework that can support different applications (touristic guiding services for museums, visiting paths recommendation for old town centers and archeological sites, etc.).

In other words, it is the user with his/her preferences (in the pre-filtering stage) and actions (in the post-filtering stage) to drive the recommendation process towards the real useful items among those that a social community considers the "best ones" (computed in the ranking stage), as in a collaborative filtering approach, where a user "learns by the others" the item utility, on the base of an *influence* measure.

The paper is organized as follows. Section 2 provides a functional overview of our system and describes the proposed strategy for recommendation. Section 3 illustrates a system customization for a tourist multimedia guide, reporting some implementation details. Section 4 reports preliminary experimental results, and provides a comparison with other recommendation techniques. Finally, Section 5 gives some concluding remarks and discusses future work.

2 The framework

2.1 System Overview

Figure 1 describes at a glance an overview of the proposed system.

Multimedia data to be recommended are retrieved by a *Wrapper* component that is composed by several modules. The *Crawler* is responsible of: (i) periodically accessing to the items' repositories (e.g., *Instagram*, *Flickr*, *Panoramio*, *Google Images*, *YouTube*,*Facebook*, *DBpedia* etc.), (ii) extracting for each item all the *features* (e.g., metadata, multimedia descriptions, etc.) and other information (e.g. user preferences, comments, time-stamped items' observations and all the different interactions between users and objects). A part of such information will be then exploited by the *Hypergraph Learning* module to build the *hypergraph* modeling the entire *Multimedia Social Network* (MSN)[6]. After the wrapping phase, all the information are stored in the *Knowledge Base* of the

2

system. In particular, it is composed by: (i) the *Multimedia Social Network Hypergraph*, (ii) the *Items DB* containing items with all the related features, (iii) *User Profiles* containing user preferences, (iv) *Contextual Data* containing some additional context information (e.g. user location, weather conditions, etc.).

Multimedia items are then grouped by the related *cultural Points Of Interest* (POIs): e.g. paintings of museum rooms, buildings in ancient ruins or in an old town center, etc.

The *Recommender Engine* provides a set of recommendation facilities for multi-dimensional and interactive browsing of items. Exploiting user preferences, the *Prefiltering* module selects a set of *candidate* items for recommendation; successively, the *Objects Ranking* module assigns a ranking of such candidates exploiting some *ranking functions* defined on the MSN. Finally, the *Postfiltering* module dynamically selects on the base of some constraints (e.g. the item that a user is currently watching and context information) a subset of candidates.

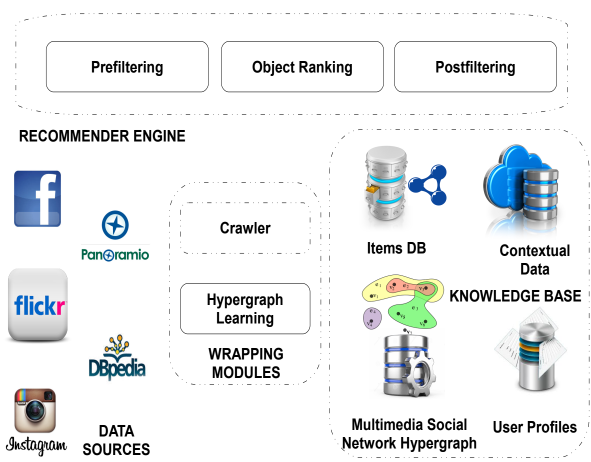


Figure 1. System Overview.

2.2 Recommendation Process

2.2.1 Pre-filtering Stage using user preferences

In the *pre-filtering* stage, our aim is to select for a given user u_h a subset $O_h^c \subset O$ containing items that are good "candidates" to be recommended.

Each item subjected to recommendation may be represented in different and heterogeneous feature spaces. For instance, a picture may be described by a set of metadata as title, description, tags, by the position in which was token and so on. Each of these sets of features contributes to the characterization of the items to different *extents*.

The first step consists in clustering together "similar" items, where the similarity should consider all (or subsets of) the different spaces of features. To this purpose, we employ *high-order star-structured co-clustering* techniques - that some of the authors have adopted in previous work [14, 7, 8] - to address the problem of heterogeneous data pre-filtering.

Let $O = \{o_1, \ldots, o_n\}$ be the set of items and $\mathcal{F} = \{F^1, \ldots, F^l\}$ a set of l feature spaces. In our recommendation problem, a user u_h is represented as a set of vectors in the same l feature spaces describing the items. To provide a first candidate list of items to be recommended, we measure the *cosine distance* of the user vectors associated to the k-th space, with the centroids of each item clusters in the k-th space. For each space, the most similar item cluster is chosen leading to l clusters $\{X_1^c, \ldots, X_l^c\}$ of candidate items.

Then, two different strategies can be adopted to provide the pre-filtered list of candidate items O_h^c: (i) *set-union strategy* - the items belonging to the union of all clusters are retained, i.e., $O_h^c = \bigcup_k X_k^c$; (ii) *threshold strategy* - the items that appears in at least ths clusters ($ths \in \{1 \ldots l\}$) are retained.

2.2.2 Ranking Stage via hypergraph modeling

The main goal of this stage is to automatically rank the set of items O embedding in a collaborative learning context: the MSN deriving by the integration of the different multimedia data sources. In particular, we use a novel technique that the authors have proposed in a previous work [6]. In our vision, a MSN is basically composed by three different kinds of entities (nodes):

- *Users* - the set of persons and organizations constituting the particular social community: several information concerning their profile, interests, preferences, etc. can eventually be considered and exploited by our model;
- *Multimedia Objects* - the set of multimedia resources (i.e. images, video, audio, posts, documents, etc.) that can be shared within a MSN community: high level (*metadata*) and low level information (*features*) can be properly used in our model;
- *Annotation Assets* - each set of symbols (e.g., keyword, tag, label, etc.) exploited by users to annotate multimedia resources within a MSN; we explicitly note that it is possible to relate a given as-

3

set with a specific *concept* (as an example a topic, a named entity, etc. which definition can be found into dictionaries, ontologies and so on), thus formally providing the related semantics.

Several types of relationships can be established among the described entities: a user can annotate an object with a particular tag, two friends can comment the same object, a user can tag another user in a photo, a user can share an object within a group, etc. In particular, we distinguish between *user to user relationships*, describing user actions towards other users, and *user to multimedia relationships*, describing user actions on objects, eventually involving some annotation assets. In addition, *similarity relationships* can be added between two objects (using multimedia features) or between two assets (by taxonomic distances).

Due to the variety and complexity of these relationships, we leverage the *hypergraph* formalism to model a MSN (all the details are provided in [6]). Then, we introduce some functions can be profitably used to "rank" users or multimedia objects in a MSN.

In our model the concept of *rank* of a given node is related to the concept of *influence*, and in our vision it can be measured by the number of user nodes that are "reachable" within a certain number of steps using any hyperpath, with respect to a social community of users, and eventually to a given *topic* of interest.

The final goal is to compute the ranking of the multimedia items in O_h^c, using as measure the social influence of each object withih the users' community.

2.2.3 Post-Filtering Stage by context information

In this stage, we have introduced a *post-filtering* method for generating the final set of "real" candidates for recommendation using *context* information.

The context is represented by means of the well-known *key-value* model [1] using as dimensions some of the different feature spaces related to items. In our system, context features can be expressed either directly using some *target items* (e.g. objects that have positively captured user attention) or specifying the related values in the shape of *constraints* that recommended items have to satisfy.

Assume that a user u_h is currently interested in a target item o_j. We can define the set of candidate recommendations as follows:

$$O_{h,j}^f = \bigcup_{k=1}^{M} \{o_i \in O_h^c \,|\, a_{ij}^k > 0\} \cup \{o_i \in NNQ(o_j, O_h^c)\} \quad (1)$$

The set of candidates includes the items that have been accessed by at least one user within k steps from o_j, with k between 1 and M, and the items that are most similar to o_j according to the results of a *Nearest Neighbor Query* ($NNQ(o_j, O_h^c)$) functionality. Note that a positive element a_{ij}^k of A^k indicates that o_i was accessed exactly k steps after o_j at least once. The ranked list of recommendations is then generated by ranking the items in $O_{h,j}^f$, for each item o_j selected as interesting by user u_h, using the ranking vector R_h thus obtaining the final set O_h^f.

Finally, for each user all the items that do not respect possible context connstraints are removed from the final list.

3 A Case Study

We have opportunely customized our system in order to provide touristic multimedia guiding services for users that are interested in visiting the old town center of Naples, Italy. On the base of user preferences and actual position, a set of POIs are shown on a proper map to tourists correlated with a multimedia description.

For instance, when a user is approaching a particular cultural POI (e.g. *Piazza del Gesu' Nuovo*), the related multimedia description and the set of candidate objects (i.e. images and texts related to the near POIs) are delivered on the user's mobile device (pre-filtering stage).

The list of proposed objects depends on the user's preferences (e.g. the majority of items will be images if a user prefers to see such kinds of data and will reveal effective user needs), is initially ordered according to effective user location (i.e. the closest items will appear at the top of list) and contains data grouped by the related cultural POI. Successively, after the user has selected one or more objects (for example the item he is currently watching), the recommendation services first perform a ranking (ranking stage) of all the candidate objects according to their recommendation grades and then filters the recommendation list considering only the most similar items to target objects (post-filtering stage).

When a user is near to a different POI, he/she can decide to modify the list of target objects (e.g. removing those related to the previous visited POI or adding new objects) and consequently recommendations will be automatically updated, thus including new items.

The design choices are briefly reported in the following.

- We consider as data source *Flickr*, *Instagram*, *Panoramio*, *DBPedia* and other domain digital

4

libraries, collecting about 500,000 items (images and texts related to historical buildings, churches, famous square and other attractions) and about 5,000 user profiles.

- As items' metadata, we consider for each multimedia item information related to *title*, *description*, *type*, *kind*, *language*, *tags*, *keyword*, *comments*, *ratings* (and for pictures the geographic position in which were taken). In addition, images are also described by a set of low-level features (i.e. SURF).

- For each item, available users' preferences, comments, feedbacks and other actions have been captured, also exploiting correlated public information from Social Networks (i.e. *Facebook*).

For what implementation details concern, the Wrapping modules leverage proper API and JAVA libraries to collect the different information of interest.

The Knowledge Base, realized using different technologies, allows to manage all the different kind of information: Contextual Data instances (messages containing information about users' position) are managed by the *Cassandra* DBMS, Items' descriptions are stored in the *Turtle* format and managed by the *AllegroGraph* repository (semantics of data can be specified by linking values of some attributes to some available ontological schema), User Profiles and the MSN hypergraph are respectively managed by *MongoDB* and *Neo4j* DBMSs.

On the other hand, the Recommender Engine exploits proper JAVA libraries (some developed for the system presented in [5] and integrated with coclustering libraries [7] and the rank refining procedure[1]) to accomplish its tasks.

Finally, a user can interact with our system using at the moment an Android Multimedia Guide App exploiting *Google Map* API.

4 Experimental Results

Recommender Systems are very complex applications that are based on a combination of several models, algorithms and heuristics. This complexity makes evaluation efforts very difficult and thus results are hardly generalizable, as reported in the literature [3]. Moreover, characterizing and evaluating the quality of a user's experience and subjective attitude toward the acceptance of recommender technology is an important issue which we will consider in the following.

[1]we use *LIRE* for the content-based image retrieval

The majority of research efforts on recommender system evaluation have mainly focused on prediction *accuracy* and *stability* (e.g., [3]).

More recently, researchers began examining issues related to users subjective opinions and developing additional criteria to evaluate recommender systems. In particular, they suggest that user satisfaction does not always (or, at least, not only) correlate with the overall recommenders accuracy.

Starting from these considerations and based on current trends in the literature, we decided to perform a *user-centric* evaluation based on *user satisfaction* with respect to assigned activities, evaluating how our recommendations can effectively support browsing tasks of different complexity when the complexity of desired itesm increases.

As in our previous work [4, 5, 7, 8], we evaluate the impact of the proposed system on users engaged in several *search tasks* of multimedia items and compared its performances with the well-known *Panoramio* system[2] that, in turn, provides basic search mechanisms.

In particular, our goal was to establish how helpful our system is in assisting the search of specific multimedia objects (images) and guiding the users towards information which satisfy their interests. The dataset used in these experiments is a subset of about 10,000 items related to specific POIs.

In order to evaluate the impact of the system on the users, we have conducted the following experiments. The system was make available to a set of 50 users. These users were all interested in the cultural heritage domain, they already had experience in the use of PCs and electronic devices, even if they were not experts in ICT. We asked these users to browse the collection of items and complete several search tasks (20 tasks per user) of different complexity (five tasks for each complexity level), using Panoramio facilities. After this test, we asked them to browse the same collection with the assistance of our recommender system and complete other 20 tasks of similar complexity. We have subdivided browsing tasks in the following four broad categories:

1. ***Low Complexity*** search tasks (T_1): e.g. find at least 30 images related to 3 different POIs depicting ancient churches;

2. ***Medium Complexity*** search tasks (T_2): e.g. find at least 50 images related to 5 different POIs depicting ancient churches, historical building and famous squares (10 objects for each subject);

[2]http://www.panoramio.com/

5

Table 1. Comparison between our system and Panoramio in terms of t_a and n_c average values

Task Class	System	$t_a(sec)$	n_c
Low Complexity	Recommender	155	38
Low Complexity	Panoramio	164	42
Medium Compl.	Recommender	335	84
Medium Compl.	Panoramio	402	104
High Complexity	Recommender	1156	298
High Complexity	Panoramio	1302	334
Very High Compl.	Recommender	1645	351
Very High Compl.	Panoramio	1832	410

3. ***High Complexity*** search tasks (T_2): e.g. find at least 100 images related to 10 different POIs (near to the actual user position) depicting ancient churches, historical building and famous squares (10 objects for each subject);

4. ***Very High Complexity*** search tasks (T_2): e.g. find at least 150 images objects related to 10 different POIs (near to the actual user position) depicting ancient churches, historical building and famous squares (15 objects for each subject).

Note that the complexity of a task depends on several factors: the number of items to explore, the type of desired features and the number of additional constraints. Two strategies were used to evaluate the results of this experiment: (i) empirical measurements of access complexity in terms of *mouse clicks* and *time*; (ii) TLX (*NASA Task Load Index factor*).

With respect to the first strategy, we measured the following parameters: (i) *access time (t_a)* – the average time spent by the users to request and access all the images for a given class of tasks; (ii) *number of clicks (n_c)* – the average number of clicks necessary to collect all the requested images for a given class of tasks.

Table 1 reports the average values of t_a and n_c for both Panoramio and our system (Recommender), for each of the four task complexity levels defined. Especially for the most complex tasks, our system shows better performances than Panormaio, especially for the more complex tasks.

We then asked the same group of users to express their opinion about the capability of Panoramio and our system respectively to provide an effective user experience in completing the assigned search tasks, based on the TLX evaluation protocol [12].

Specifically, TLX is a multi-dimensional rating procedure that provides an overall workload score based on a weighted average of ratings on six sub-scales: mental demand, physical demand, temporal demand, own performance, effort and frustration. Lower TLX scores are better and the average scores are then reported in Table 2.

Table 2. Comparison between our system and Panoramio in terms of TLX factors for each category of users

TLX factor	Recommender	Panoramio
Mental demand	39	41
Physical demand	36.3	48
Temporal demand	39	50
Effort	35	50.5
Perfomances	69.7	79.8
Frustation	33.2	44.1

Our system outperforms in a significative way Panormaio in every sub-scale except for *mental demand* and *performance*: this happens because sometimes an expert user considers the automatic suggestions not useful, just because they know what they are looking for.

In summary, our system provides a better (less frustrating) user experience during the search tasks. In addition, the fact that search tasks can be completed faster using our system is an indication that recommendations are effective, as they allow a user to explore interesting and related items one after another, without the interference of undesired items that would otherwise slow down the process.

5 Conclusions and Future Work

In this paper a novel multimedia and scoial recommendation approach for Cultural Heritage applications. It combines several aspects of users - i.e. their *preferences* (usually in the shape of items' metadata) and *interactions* within a social community modeled using hypergraphs - together with items' *multimedia features* and *context information* within a general framework that can support different applications (touristic guiding services for museums, visiting paths recommendation for old town centers and archeological sites, etc.).

Preliminary experiments on user satisfaction demonstrated how our approach achieve very promising and interesting results. Future works will be devoted to extend the experimental evaluation to a larger multimedia data set, also considering the performance, evaluated in terms of accuracy, precision and recall, of the performed recommendations. Moreover, we plan to apply our approach to other kinds of data gathered from heterogeneous collections and compare our approach with other ones proposed in the literature.

References

[1] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, 23(1):103–145, 2005.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17:734–749, 2005.

[3] Gediminas Adomavicius and Jingjing Zhang. Stability of recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 30(4):23, 2012.

[4] Massimiliano Albanese, Antonio d'Acierno, Vincenzo Moscato, Fabio Persia, and Antonio Picariello. A multimedia semantic recommender system for cultural heritage applications. In *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, pages 403–410. IEEE, 2011.

[5] Massimiliano Albanese, Antonio dAcierno, Vincenzo Moscato, Fabio Persia, and Antonio Picariello. A multimedia recommender system. *ACM Transactions on Internet Technology (TOIT)*, 13(1):3, 2013.

[6] Flora Amato, Vincenzo Moscato, Antonio Picariello, and Giancarlo Sperli'. Multimedia social network modeling: a proposal. In *Semantic Computing (ICSC), 2016 10th IEEE International Conference on*. IEEE, 2016.

[7] Ilaria Bartolini, Vincenzo Moscato, Ruggero G Pensa, Antonio Penta, Antonio Picariello, Carlo Sansone, and Maria Luisa Sapino. Recommending multimedia visiting paths in cultural heritage applications. *Multimedia Tools and Applications*, pages 1–30, 2014.

[8] Francesco Colace, Massimo De Santo, Luca Greco, Vincenzo Moscato, and Antonio Picariello. A collaborative user-centered framework for recommending items in online social networks. *Computers in Human Behavior*, 51:694–704, 2015.

[9] Ruihai Dong, Michael P. O'Mahony, Markus Schaal, Kevin McCarthy, and Barry Smyth. Sentimental product recommendation. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 411–414, New York, NY, USA, 2013. ACM.

[10] Paul Dourish. What we talk about when we talk about context. *Personal and ubiquitous computing*, 8(1):19–30, 2004.

[11] Gayatree Ganu, Yogesh Kakodkar, and AméLie Marian. Improving the quality of predictions using textual information in online user reviews. *Inf. Syst.*, 38(1):1–15, March 2013.

[12] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, 52:139–183, 1988.

[13] Yoshinori Hijikata, Kazuhiro Iwahama, and Shogo Nishida. Content-based music filtering system with editable user profile. In *Proceedings of the 2006 ACM symposium on Applied computing*, SAC '06, pages 1050–1057, New York, NY, USA, 2006. ACM.

[14] Dino Ienco, Céline Robardet, Ruggero G Pensa, and Rosa Meo. Parameter-less co-clustering for star-structured heterogeneous data. *Data Mining and Knowledge Discovery*, 26(2):217–254, 2013.

[15] Katerina Kabassi. Personalisation systems for cultural tourism. In *Multimedia services in intelligent environments*, pages 101–111. Springer, 2013.

[16] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86. ACM, 2010.

[17] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.

7

[18] Cane WK Leung, Stephen CF Chan, and Fu-lai Chung. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In *Proceedings of The ECAI 2006 Workshop on Recommender Systems*, pages 62–66. Citeseer, 2006.

[19] Veronica Maidel, Peretz Shoval, Bracha Shapira, and Meirav Taieb-Maimon. Evaluation of an ontology-content based filtering method for a personalized newspaper. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 91–98, New York, NY, USA, 2008. ACM.

[20] Katarzyna Musial, Krzysztof Juszczyszyn, and Przemyslaw Kazienko. Ontology-based recommendation in multimedia sharing systems. *System Science*, 34:97–106, 2008.

[21] Nikolaos Pappas and Andrei Popescu-Belis. Sentiment analysis of user comments for one-class collaborative filtering over ted talks. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 773–776, New York, NY, USA, 2013. ACM.

[22] Michael Pazzani and Daniel Billsus. Content-Based Recommendation Systems. pages 325–341. 2007.

[23] Naren Ramakrishnan, Benjamin J. Keller, Batul J. Mirza, Ananth Y. Grama, and George Karypis. Privacy risks in recommender systems. *IEEE Internet Computing*, 5:54–62, November 2001.

[24] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. pages 175–186. ACM Press, 1994.

[25] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[26] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 253–260, New York, NY, USA, 2002. ACM.

[27] Vivek Kumar Singh, Mousumi Mukherjee, and Ghanshyam Kumar Mehta. Combining collaborative filtering and sentiment classification for improved movie recommendations. In *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, pages 38–50. Springer, 2011.

[28] Johann Stan, Fabrice Muhlenbach, Christine Largeron, et al. Recommender systems using social network analysis: Challenges and future trends. *Encyclopedia of Social Network Analysis and Mining*, pages 1–22, 2014.

[29] Xiaoyuan Su and Taghi Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, 2009.

[30] Hilmi Yildirim and Mukkai S. Krishnamoorthy. A random walk method for alleviating the sparsity problem in collaborative filtering. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 131–138, New York, NY, USA, 2008. ACM.

[31] Hsiang-Fu Yu, Cho-Jui Hsieh, Si Si, and Inderjit S Dhillon. Parallel matrix factorization for recommender systems. *Knowledge and Information Systems*, pages 1–27, 2013.

[32] Xujuan Zhou, Yue Xu, Yuefeng Li, Audun Josang, and Clive Cox. The state-of-the-art in personalized recommender systems for social networking. *Artif. Intell. Rev.*, 37(2):119–132, February 2012.

8

VisCFSM: Visual, Constraint-Based, Frequent Subgraph Mining

Jennifer L. Leopold
Missouri University of Science & Technology
Department of Computer Science
Rolla, Mo USA
leopoldj@mst.edu

Chaman L. Sabharwal
Missouri University of Science & Technology
Department of Computer Science
Rolla, MO USA
chaman@mst.edu

Nathan W. Eloe
Northwest Missouri State University
School of Computer Science and Information Systems
Maryville, MO USA
nathane@nwmissouri.edu

***Abstract*—Graphs long have been valued as a pictorial way of representing relationships between entities. Contemporary applications use graphs to model social networks, protein interactions, chemical structures, and a variety of other systems. In many cases, it is useful to detect patterns within graphs. For example, one could be interested in identifying frequently occurring subgraphs, which is known as the frequent subgraph mining problem. A complete solution to this problem can result in numerous subgraphs and can be time-consuming to compute. An approximate solution is faster, but is subject to static heuristics that are beyond the control of the user. Herein we present VisCFSM, a visual, constraint-based, frequent subgraph mining system which allows the user to dynamically specify a variety of constraints on the subgraphs to be found while the mining algorithm is running. The constraint specification interactions are performed through a visual user interface, thereby facilitating a form of visual algorithm steering. This approach can be integrated with any frequent subgraph mining algorithm. Most importantly, this approach has the potential for the user to better, and more quickly, find the information that is of most interest to him/her in a graph.**

Keywords-graph; data mining; visual algorithm steering

I. INTRODUCTION

Graphs long have been valued as a pictorial way of representing complex relationships between entities. Commercial, research, and government organizations use graphs to model social networks, protein interactions, chemical structures, and a variety of other systems. A common application of graph data mining is to identify the most recurrent relationships or patterns amongst the data in a graph, which typically requires finding frequently occurring subgraphs.

For some applications, the input will be a collection of relatively small graphs, and the search for frequent subgraphs is performed over each individual graph in the collection before those results are combined. This is known as a *graph-transaction setting*. In contrast, the input may be a single graph; this is referred to as a *single graph setting*. Our work refers to the latter environment. We also restrict our work to *static graphs*, and do not address *dynamic graphs* or *streaming graphs*, which are discussed in [1].

DOI reference number: 10.18293/DMS2016-026

Formally, we define the Frequent Subgraph Mining (FSM) problem as in the paper by Abedijaberi [2] using Definitions 1-4 given below.

Definition 1. *A labelled graph $G = (V, E, L_V, L_E)$ consists of a set of vertices V, a set of undirected or directed edges E, and two labeling functions L_V and L_E that association labels with vertices and edges, respectively*.

It should be noted that the labels of any two vertices (or any two edges) may not be unique. However, each vertex (and each edge) will have a unique *id*.

Definition 2. *A graph $S = (V_S, E_S, L_{VS}, L_{ES})$ is a subgraph of $G = (V, E, L_V, L_E)$ iff $V_S \subseteq V$, $E_S \subseteq E$, $L_{VS}(v) = L_V(v)$ and $L_{ES}(e) = L_E(e)$ for all $v \in V_S$ and $e \in E_S$.*

Definition 3. *A subgraph isomorphism of S to G is a one-to-one function $f: V_S \rightarrow V$ where $L_{VS}(v) = L_V(f(v))$ for all vertices in $v \in V_S$, and for all edges $(u,v) \in E_S$, $((f(u),f(v)) \in E$ and $L_{ES}(u,v) = L_E((f(u), f(v))$.*

Definition 4. *Let I_S be the set of isomorphisms of a subgraph S in graph G. Given a minimum support threshold τ, the frequent subgraph mining problem (FSM) is to find all subgraphs S in G such that $|I_S| \geq \tau$.*

The advantage of limiting frequent subgraphs to only those with disjoint edges is computational tractability [3]. But this comes at the expense of disregarding potentially useful information. Hence, in our work we allow isomorphic subgraphs to share edges.

FSM algorithms that find complete solutions may, depending upon the specified threshold value and the size of the graph, result in numerous subgraphs and take a considerable amount of time to compute. Algorithms that find approximate solutions are faster, but apply static heuristics that are beyond the control of the user (unless s/he modifies the software).

Herein we present VisCFSM, a visual, constraint-based, frequent subgraph mining system which allows the user to dynamically specify a variety of constraints on the subgraph mining algorithm while it is running. The constraint specification interactions are performed through a visual user interface, thereby facilitating a form of visual algorithm steering.

The prototype implementation uses the FSG [4] frequent subgraph algorithm; however, the approach we employ can be integrated with any FSM algorithm. Most importantly, this approach has the potential for the user to better, and more quickly, find the information that is of most interest to him/her in a graph.

The organization of this paper is as follows. Section II provides a brief overview of related work in graph data mining. Motivation for the need for dynamic, visual steering of FSM using constraints is presented in Section III. In Section IV, we discuss the VisCFSM infrastructure in terms of the FSM, the constraint satisfaction system, and the graphical user interface. An example of running VisCFSM is presented in Section V. Finally, we discuss our plans for future work in Section VI and conclusions in Section VII.

II. RELATED WORK

A. *Graph Data Mining Algorithms*

Graph Data Mining (GDM) algorithms are divided into three main categories: Graph Theory Based, Inductive Logic Programming, and Greedy Search [5]. Our work focuses on the Graph Theory Based category, which consists of two main groups: Apriori-based and pattern growth-based approaches. Algorithms in the first group generate candidate subgraphs by joining two frequent subgraphs of the same size to generate larger subgraphs. Pattern growth algorithms generate candidates by adding a new edge to each smaller frequent subgraph.

FSM algorithms typically face two computational challenges: (i) candidate subgraph generation, and (ii) identification of candidate subgraphs that meet the minimum support threshold. In the worst case, all subgraphs in the graph must be examined, which is exponential in complexity, and subgraph isomorphisms must be computed, which is an NP-complete problem. FSM algorithms may attempt to improve runtime performance by reducing the size of the search space, avoiding duplicate comparisons, and/or minimizing the amount of memory required for compiling intermediate results. Another solution to reduce the runtime is to provide an approximate, rather than a complete, solution to the FSM problem.

B. *Heuristics for Approximate Solutions*

Heuristic FSM algorithms such as SUBDUE [6], GREW [7] and GRAMI [8] discover only a subset of all frequent subgraphs of a graph. These algorithms do not return any infrequent patterns (i.e., the results do not have false positives), but may miss some frequent ones (i.e., the results effectively may have false negatives). The type of heuristics that are employed are quite diverse, and also vary considerably in their degree of complexity. Some examples are listed below:

- SUBDUE [6] starts with frequent subgraphs consisting of a single vertex, and then expands those in a breadth-first manner by adding a new edge. The order of processing is known as a “beam search”, and only a predetermined number of paths (i.e., the beam width) are kept as candidates at each iteration. Hence some valid frequent subgraphs will be missed.
- Like SUBDUE, GREW [7] employs a beam search to prune large portions of the search space. It also iteratively joins frequently occurring pairs of nodes into a single supernode, and determines disjoint embeddings of connected subgraphs using a maximal independent set algorithm. GREW employs an additional heuristic that deliberately underestimates the frequency of each discovered subgraph in an attempt to reduce the search space. While experiments showed that GREW significantly outperformed SUBDUE with respect to runtime, those experiments showed that this came at the expense of finding fewer frequent subgraphs.
- Pattern growth algorithms generate candidate subgraphs by adding a new edge to smaller frequent subgraphs. GRAMI [8] only adds frequently occurring edges to smaller frequent subgraphs when generating candidate subgraphs. This will miss finding some valid frequent subgraphs, but reduces the total number of iterations over edges that must be considered.
- AGRAMI [8] is an extension of GRAMI that employs additional heuristics in an effort to scale to larger graphs. For example, it enforces a timeout when testing whether a subgraph occurs at least as many times as the minimum support threshold; if the solution cannot be computed within a particular amount of time, that subgraph is assumed to be infrequent.

In the same paper that presents GRAMI and AGRAMI [8], the authors briefly discuss CGRAMI, a version of GRAMI that seeks to find more general patterns in graphs than just frequent subgraphs. This work is noteworthy to mention herein because it claims to support the following user-defined constraints:

- Number of vertices (or edges) in a pattern cannot exceed a specified value
- Vertex degree in a pattern cannot exceed a certain value
- A pattern must include/exclude only vertices with certain labels
- A pattern must include only certain edges
- A pattern cannot include certain edges
- A pattern cannot include a specified subgraph
- A specified vertex label cannot appear more than N times in a pattern

To specify desired constraints in CGRAMI, the user must comment out certain lines of code (and uncomment other lines) for the constraints, set the values for parameters, and then recompile the program. The program has a command-line interface; there is no graphical user interface.

As stated previously, what all heuristic FSM (and constraint-based GDM) algorithms have in common is the inability for the user to dynamically customize the heuristics, or any form of constraints, while the algorithm is running. This is the novel contribution of the work presented herein.

III. MOTIVATION

In part, motivation for this work came from a graduate course on Advanced Data Mining taught by author Leopold in 2015 at Missouri University of Science and Technology. That year the focus of the course was graph data mining. Students read several research papers on GDM algorithms and applications. Some of the students in the class implemented a few of the algorithms in Python, but were frustrated that they had to wait a considerable amount of time for the computation on some relatively small graphs (e.g., a graph of 50 vertices with average vertex degree 3.5 took over 8 hours to compute all frequent subgraphs of minimum support 2). When they had their programs output intermediate results as the subgraphs were being found, sometimes the students would terminate the program, and restart it with a different threshold value to further discriminate the result set and make the program finish more quickly.

At the end of the course, the students were asked what kinds of constraints they would have found useful to "steer" a FSM algorithm dynamically, even if it meant that the resulting set of subgraphs would not be complete. Here we use the term *steering* as discussed in [9]: the ability to have a continuous visualization of the (output) data as a program executes, coupled with the ability for the programmer to interactively modify any aspect of the program and see the effects without restarting the computation.

With a social network (specifically, a terrorist network) as an application domain, the students identified the constraints and use case examples listed below. In this social network, it is assumed that a vertex in the graph is labelled with a person's name, which are not necessarily unique. Additional information about a person and his/her relationship to other people may be represented in the graph as vertex or edge data.

- **Include/exclude subgraphs containing a certain set of vertices.** Ex.: Suppose that we're using a social media network to identify terrorist threats. The number of frequent subgraph results may be quite high at first due to very small terrorist groups. So we then want to narrow our search, and only continue to look for subgraphs that include a specific group of people that we know conduct terrorist activities.
- **Include/exclude only frequent subgraphs that appear more/less than subgraph (or vertex) X does.** Ex.: We see a specific name in the preliminary results of our search that we already know is a leader and a threat. But his name isn't the only one we see, and we want to know who in the group is more important, of high rank, or higher rank than this person. So we then narrow our search to find subgraphs that appear more often than those containing this person. Or maybe we are looking for someone we can capture and get information from, in which case we look for someone important who appears less often.
- **Include/exclude only frequent subgraphs that are disconnected/connected to subgraph (or vertex) X.** Ex.: We begin a search on terrorist cells. However, based on seeing a particular group appearing frequently in the results, we want to narrow our search to those connected to that group. Similarly, if we are trying to identify new terrorist cells or rival cells, we may want to look only at those groups that are disconnected from a certain group.
- **Include/exclude only frequent subgraphs where the average vertex degree is greater than some number.** Ex.: We're looking for potential terrorist cells, and not interested in groups with only a couple of connections; such groups are unlikely to be funded or be a real threat. We may not see this until after we have seen the initial (small-sized) frequent subgraphs.
- **Include/exclude subgraphs containing a certain number of edges.** We may not be interested in seeing small terrorist groups, but rather want to see a certain amount of interconnectivity; these might prove to be the more dangerous terrorist groups.
- **Change minimum support.** Ex.: We may start our search very wide open, but, after seeing some preliminary results that are too numerous and/or contain trivial information (e.g., everyone is a potential terrorist), decide that we want to raise the threshold.

It should be noted that these constraints are not intended to be mutually exclusive, but rather conjunctive; we should be able to specify any combination of constraints.

In the next section, we discuss the VisCFSM infrastructure in terms of the FSM, the constraint satisfaction system, and the graphical user interface. The system was designed to address many of the above listed constraints.

IV. VISCFSM

The infrastructure of VisCFSM consists of a front end and a back end. The front end is comprised of the graphical user interface which displays the frequent subgraphs as they are computed, and allows the user to visually steer the FSM by specifying constraints on frequent subgraph selection as the algorithm is progressing. The back end consists of the FSM and the constraint satisfaction system. In this section we briefly discuss each part of the infrastructure.

A. The FSM

As mentioned in Section II, we have chosen to focus on the Graph Theory Based category of graph data mining algorithms, which consists of Apriori-based and pattern growth-based approaches. For the prototype implementation of VisCFSM we chose a pattern growth algorithm, FSG [4]. The algorithm starts by finding all frequent subgraphs consisting of one edge. It then makes repeated iterations, generating candidates by adding a new edge to each of the largest frequent subgraphs found so far. This particular algorithm was selected primarily for its simplicity; it is certainly not one of the most efficient FSM algorithms that exists, but we believed that the logic upon which it is based could easily be understood by most users. The choice of FSM algorithms to be used in VisCFSM is not important; the constraint satisfaction system and visualization control system that we employ actually can be integrated with any FSM algorithm.

B. The Constraint Satisfaction System

Inspired by the use cases presented in Section III, several structural and semantic constraints have been implemented for VisCFSM. These are listed below:

- Include/exclude frequent subgraphs that contain certain vertices or edges
- Include/exclude frequent subgraphs that include a particular subgraph
- Include subgraphs that are connected/disconnected to a particular vertex or edge
- Include only frequent subgraphs that have at least one vertex that has degree greater than a specified number
- Include only frequent subgraphs where the average vertex degree greater than a specified number
- Include/exclude frequent subgraphs containing a certain number of edges
- Exclude frequent subgraphs where a certain vertex label appears greater than a specified number of times
- Change minimum support

The user interface allows the user to specify the constraints that should be applied to the set of frequent subgraphs found so far, and whether to continue applying these constraints in the next iteration of the algorithm in an effort to find new frequent subgraphs (e.g., in the case of the FSG algorithm, the next iteration adds an edge to each of the largest-sized frequent subgraphs found so far in order to form new candidate frequent subgraphs).

C. The Graphical User Interface

The VisCFSM FSM and constraint satisfaction system were implemented in SWI-Prolog. The choice of a logic programming language seemed most suitable for modeling a constraint satisfaction problem. However, SWI-Prolog has no graphical capabilities. Hence, the VisCFSM graphic user interface was developed in Python.

The graphic user interface (GUI) consists of the following controls: (i) a file chooser to allow the user to select a Prolog file that contains the specification of a graph, (ii) a text input field to specify the name of the graph (i.e., a Prolog file may contain multiple graph specifications, each defined as a relation), (iii) a text input field to specify the minimum support threshold for considering a subgraph to be frequent, (iv) a constraint editor, (v) a control button to start the FSG by finding the smallest-sized FSGs, and (vi) a control button to add an edge to each of the largest FSGs found thus far. A graph specification consists of a Prolog list containing the list of vertices (in the format [ID, label]) and a list of edges, where each edge is represented as a list of two vertices. Fig. 1 shows the GUI after an undirected graph named *sampleGraph* has been loaded from a Prolog file named *graph.pl*. In this figure, no frequent subgraphs have been found yet.

The constraint editor allows the user to set up rules to filter the frequent subgraphs that will be reported. Examples of the constraint editor are shown in Fig. 2 and Fig. 3. Constraints are represented in Disjunctive Normal Form (DNF); that is, as a series of AND clauses OR'd together. The editor includes a drop-down menu of the possible constraints, a text input field for specifying the arguments to a constraint, and a display of the DNF clauses that have been specified so far. Help text is also provided to guide the user in specifying the arguments correctly.

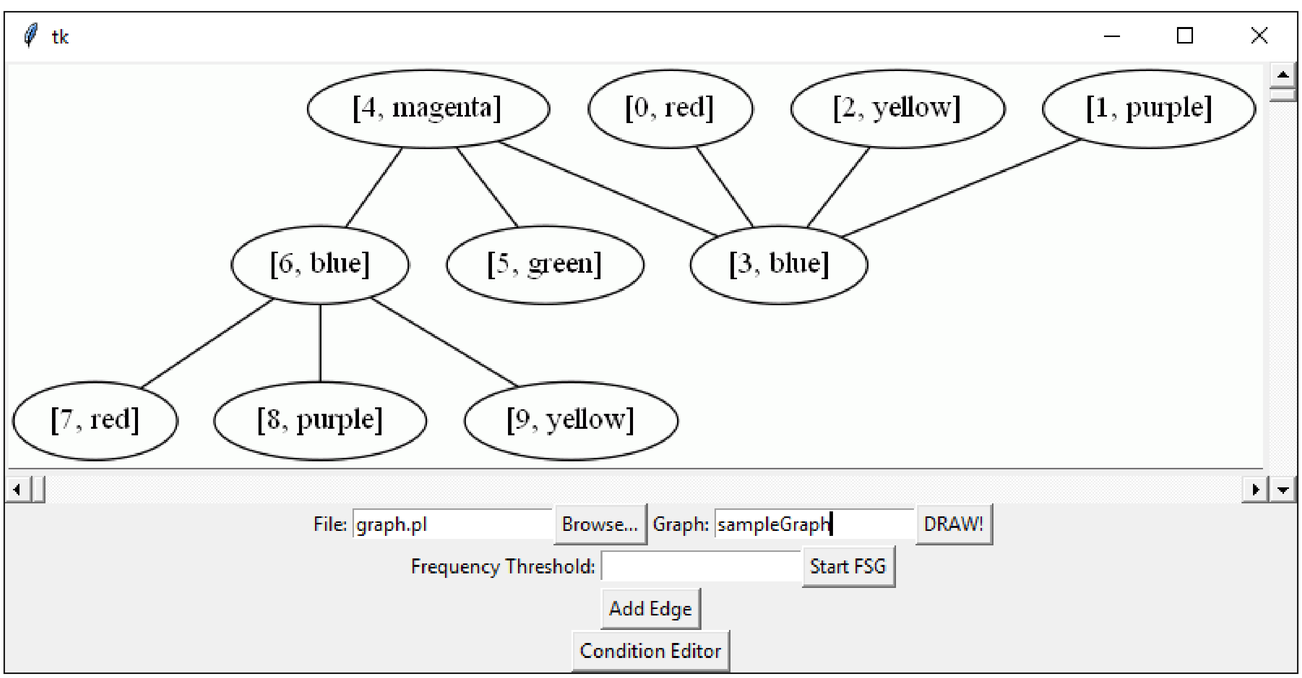


Figure 1. VisCFSM GUI after a sample graph has been loaded

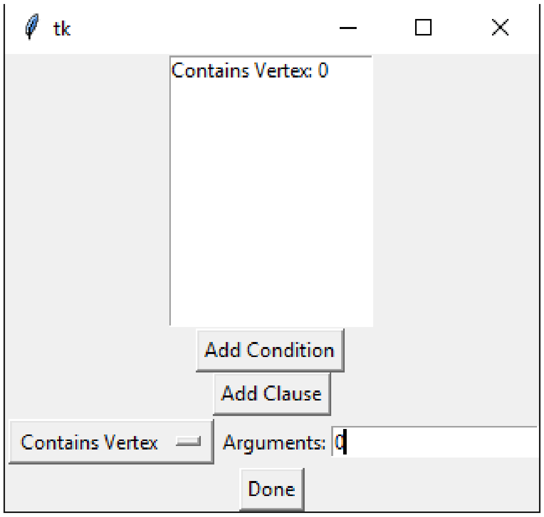


Figure 2. Constraint editor showing one clause

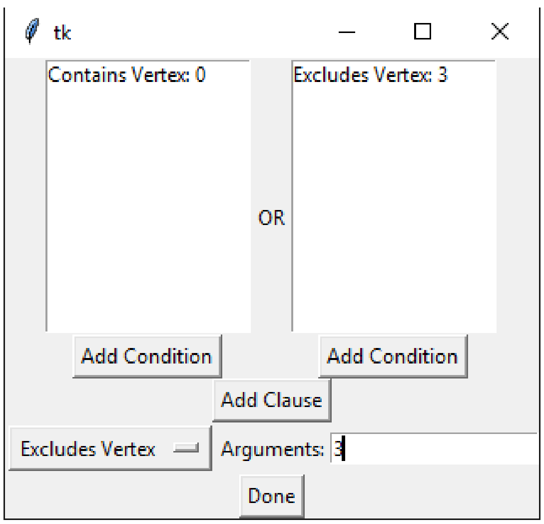


Figure 3. Constraint editor showing two clauses

The main display area in the GUI initially shows the graph that the user has specified from the selected Prolog file. Once the FSG algorithm is invoked, that area of the GUI is used to display the frequent subgraphs found in the most recent iteration of the algorithm. Recall that FSG starts by finding all frequent single-edge subgraphs, then makes repeated iterations, adding a new edge to each of the largest frequent subgraphs found so far. In future refinements of the GUI, the user will be given the ability to scroll back to previously displayed sets of (smaller-sized) frequent subgraphs, and also will be given the option to undo/redo the application of constraints and edge additions. With the current implementation, at any time, the user may restart the FSG generation algorithm from the beginning by clicking on the **Start FSG** button.

V. An Example in VisCFSM

To demonstrate the concepts behind VisCFSM, here we walk through two simple examples. We start by assuming that we have reached the state shown in Fig. 1, having specifying the file *graph.pl* and selecting *sampleGraph* as the desired graph. Clicking on the **DRAW**! button renders the graph without finding any frequent subgraphs.

For this particular graph, a minimum support threshold of 2 provides interesting results. Fig. 4 shows all frequent subgraphs with a minimum support of 2. These are listed individually with their unique vertex IDs rather than, for example, simply reporting that a subgraph with edge *(blue, red)* occurs at least 2 times; there is, however, an option in the GUI to report the results only by unique label combinations in the subgraphs.

Suppose that we want only subgraphs that contain a specific vertex, say those with the ID 0. The user would first click on the **Condition Editor** button, which will open the condition editor window. In the condition editor, the user would then select "Contains Vertex" from the dropdown menu, and specify the ID of the vertex to include; see Fig. 2. Finally, the user would click on the **Add Condition** button. The constraint editor can be closed by clicking on **Done** or closing the window.

Upon completing those steps, if the user now clicks the **Start FSG** button, the FSG algorithm will be restarted with the specified constraint applied; these results are shown in Fig. 5. At this point, the user may increment the size of these subgraphs (by clicking on the **Add Edge** button), or go back to the constraint editor, add additional constraints to the current AND clause, and recompute the set of frequent single-edge subgraphs.

To add an alternative set of constraints (i.e., add another clause that will be OR'd in the DNF representation), the user can open the constraint editor and click on the **Add Clause** button. A new clause column will be displayed, and the user now can add constraints to either of the clauses. Fig. 3 shows the addition

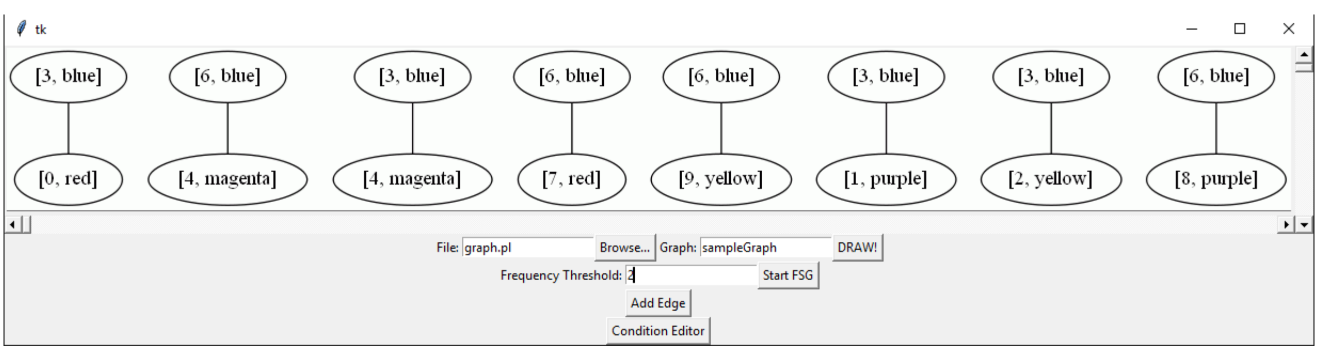


Figure 4. Subgraphs from the graph shown in Fig. 1 that occur with minimum support 2

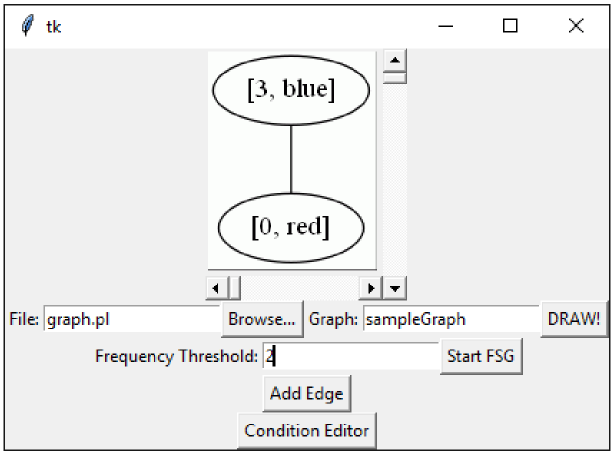


Figure 5. Results of applying the constraint specified in Fig. 2 when finding single-edge frequent subgraphs for the graph shown in Fig. 1

of another constraint in our example; namely we also want frequent subgraphs that exclude the vertex with ID 3. The resulting set of frequent single-edge subgraphs (obtained by clicking the **Start FSG** button) is shown in Fig. 6. The result set now includes single-edge subgraphs that occur at least 2 times, and include vertices with ID 0 or exclude vertices with ID 3.

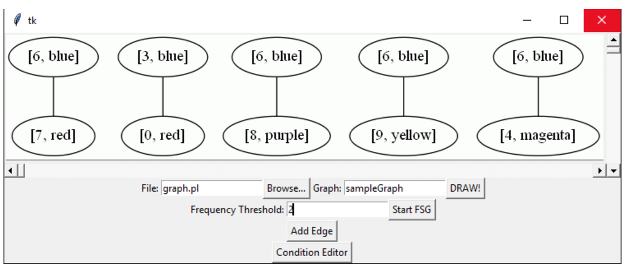


Figure 6. Results of applying the constraints specified in Fig. 3 when finding single-edge frequent subgraphs for the graph shown in Fig. 1

VI. Future Work

Future work on VisCFSM will be focused on three areas. First, we intend to perform informal studies of the usefulness and usability of the system when author Leopold again teaches the graduate course on Advanced Data Mining at Missouri University of Science and Technology in 2017. This should yield ideas for improving existing features in the user interface and adding new constraints.

Secondly, we plan to explore the incorporation of specifications of constraints by natural language and/or gestures/drawing in the user interface. The use of multimodal user interfaces for spatially-oriented applications have been of interest for years, particularly for GIS applications; see the discussion in [10], for example. Such interfaces have become a particularly active research topic in the past few years with the prevalence of mobile GIS applications (e.g., [11], [12]). Many of the same concepts for querying general spatial settings (e.g., objects such as buildings, connections between objects such as roads, directionality, etc.) should also be applicable to specifying patterns in a graph.

The third focus for future work will be improving the runtime efficiency of VisCFSM. Most algorithms used for graph data mining are designed for implementations in procedural languages. It is possible that some of our FSM operations (written in Prolog) could be sped up by expressing the steps of those algorithms (e.g., testing for graph isomorphism) in a more logical form rather than mimicking procedural solutions.

Additionally, many parts of the algorithm are embarrassingly parallel in nature; operations that are performed on every subgraph in the list of frequent subgraphs are independent from those being applied to other subgraphs, and as such can be done in a parallel manner. SWI-Prolog offers the concurrent/3 and concurrent_maplist/N predicates that automatically distribute the execution of goals across multiple threads. As threading introduces significant overhead to the process, determining the appropriate size of problems to apply concurrent operations will require empirical experimentation so as to not degrade the performance of the system.

Another option for exploring concurrency in VisCFSM includes using a Prolog interpreter embedded in the language in which the GUI is implemented. Javascript, for example, has a library that is a Prolog interpreter implemented in Javascript. This would allow the GUI to use the threading and GUI features of the host language (in Javascript's case, Electron or NW.js could provide the GUI features) while preserving the Prolog computational backend. This has the disadvantage of requiring the GUI to manage communication between threads, but the higher-level threading abilities may make this a preferable option to sending queries to a Prolog process.

If reconciling the data representations of Prolog and a host language prove to introduce more complexity than is needed, a domain-specific logic language that integrates tightly with other languages (such as miniKanren [13]) would provide a more native approach to interacting with the computational backend. This would provide access to high-level threading and GUI capabilities of more general-purpose programming languages while leveraging the constraint programming capabilities of a logic programming language.

VII. Summary and Conclusions

Herein we have presented the infrastructure for a graph mining system that provides the ability for the user to dynamically customize a variety of semantic and structural constraints while the algorithm is working to complete its overall task. Effectively, this system supports visual algorithm steering, providing the ability for the user to continuously visualize the results of the graph mining program as it executes, interactively modify aspects of the program, and see the effects without restarting the computation from the very beginning. Such capabilities are extremely valuable when dealing with graph mining, wherein the data representation is intrinsically visual, and patterns of interest may not become obvious until preliminary results are seen. Because frequent subgraph mining is a computationally intensive problem, the ability to dynamically adjust constraints on the computation can allow the user to more quickly find the information that is of most interest to him/her in a graph.

REFERENCES

[1] A. Bifet, G. Holmes, B. Bfahringer, and R. Gavald'a, "Mining frequent closed graphs on evolving data streams", Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, 2011, pp. 591-599.

[2] A. Abedijaberi and J.L. Leopold, "FSMS: a frequen subgraph mining algorithm using mapping sets," Proceedings of the 12th International Conference on Machine Learning and Data Mining, New York, NY, 2016.

[3] M. Kuramochi and G. Karypis, "Finding frequent patterns in a large sparse graph," Data Mining and Knowledge Discovery, 11.3, 2005, pp. 243-271.

[4] M. Kuramochi and G. Karypis, "Frequent subgraph discovery," Proceedings of the 2001 IEEE International Conference on Data Mining, IEEE Computer Society, 2001.

[5] M. Gholami and A. Salajegheh, "A survey on algorithms of mining frequent subgraphs," International Journal of Engineering Inventions, 1.5, 2012, pp. 60-63.

[6] D.J. Cook and L.B. Holder, "Substructure discovery using minimum description length and background knowledge, " Journal of Artificial Intelligence Research, 1(1), 1994, pp. 231-255.

[7] M. Kuramochi and G. Karypis, "An efficient algorithm for discovering frequent subgraphs," IEEE Transactions on Knowledge and Data Engineering, 16(9), September 2004, pp. 1038-1051.

[8] M. Elseidy, E. Abdelhamid, S. Skiadopoulos, and P. Kalnis, "GRAMI: frequent subgraph and pattern mining in a single large graph," Proceedings of VLDB Endowment, 2014, pp. 517-528.

[9] J.W. Atwood, Jr., M.M. Burnett, R.A. Walpole, E.M. Wilcox, and S. Yang, "Steering programs via time travel", 1996 IEEE Symposium on Visual Languages, Boulder, CO, 1996, pp. 1-8.

[10] I. Schlaisich and M.J. Egenhofer, "Multimodal spatial querying: what people sketch and talk about", 1st International Conference on Universal Access in Human-Computer Interaction, 2001, pp. 732-736.

[11] F. Cutugno, V.A. Leano, R. Rinaldi, and G. Mignini, "Multimodal framework for mobile application", Proceedings of the International Working Conference on Advanced Visual Interfaces, New York, NY, 2012, pp. 197-203.

[12] J. Doyle, M. Bertolotto, and D. Wilson, "Evaluating the benefits of multimodal interface design for CoMPASS – a mobile GIS", GeomInformatica, Volume 14, Issue 2, April 2010, pp. 135-162.

[13] C.E. Alvis, J.J. Wilcock, K.M. Carter, W.E. Byrd, and D.P. Friedman, "cKanren miniKanren with constraints", 2011.

KnotSketch: a tool for knot diagram sketching, encoding and re-generation

Gennaro Costagliola[1], Mattia De Rosa[1], Andrew Fish[2], Vittorio Fuccella[1], Rafiq Saleh[3], Sarah Swartwood[2]
[1]Dipartimento di Informatica, University of Salerno, Via Giovanni Paolo II, 84084 Fisciano (SA), Italy
[2] School of Computing, Engineering and Mathematics, University of Brighton, UK
[3]Department of Earth, Ocean and Ecological Sciences, University of Liverpool, UK
{gencos, matderosa, vfuccella}@unisa.it, {andrew.fish,S.Swartwood}@brighton.ac.uk

Abstract

Knots occur in many areas of science and art. The mathematical field of Knot Theory studies an idealised form of knots by viewing them as closed loops in 3-space. They can be formally studied via knot drawings which are well-behaved projections of the knot onto the 2-D plane. Equivalence of knots in 3-space (ambient isotopy) can be encapsulated via sequences of diagram rewriting rules, called Reidemeister moves, but finding such sequences demonstrating isotopy of two knots can be immensely challenging. Whilst there are some sophisticated tools available for some knot theoretic tasks, there is limited (free) tool support for certain knot creation and interaction tasks, which could be useful for lecturers and students within University courses. We present KnotSketch, a tool with multiple functionalities including the ability to: (i) read off a form of Gauss code for a user sketched diagram; (ii) generate a diagram from such a code; (iii) regenerate a knot diagram via a different projection, thereby producing examples of equivalent knot diagrams that may look very different; (iv) interaction capabilities to quickly alter the knot via crossing changes and smooth the curves of the sketched diagram; (v) export facilities to generate svg images of the constructed knots. We evaluate KnotSketch via a case study demonstrating examples of intended usage within an educational setting. Furthermore, we performing a preliminary user study to evaluate the general usability of the tool.
Keywords: *knots, sketching, gauss code, diagram generation.*

1. Introduction

Knots occur within both art and science, and there are many important scientific application domains (e.g. DNA supercoiling [22, 8], quantum wavefunctions [16]). The mathematical field of Knot Theory has been studied extensively, providing a rigorous study of an idealised form of knots (essentially closed loops in 3-space); see [27] for a standard mathematics graduate text book, and [19] for a recent approach aiming to utilise the potential of computers within the field. A standard mathematical approach is to define objects under consideration, provide a formal notion of equivalence, and then to investigate means to try to classify the objects (given two objects, decide if they are equivalent or inequivalent). In this context, two knots are equivalent, called ambient isotopic (isotopic for short), if there is a continuous deformation of the whole of 3-space taking one embedded loop to the other. Knots can be studied thus, making use of knowledge of topological methods.

However, knots can also be studied combinatorially, via knot diagrams (which are regular projections of knots onto a plane, so any crossings are transverse double points). Following Reidemeister [25], isotopy can be realised via diagram transformations: two knots are isotopic if and only if their diagrams (which can be projection onto any suitable plane, so the diagrams can look quite different) differ by a sequence of local diagrammatic transformations (shown later, in Figure 4). Figure 1 shows an instance of the core problem of asking whether diagrams represent the same knot; no answer is provided here, leaving the reader to try to discover an answer for themselves, thereby getting an initial feeling for the challenge via this small example. This question is posed at the beginning of an undergraduate Knot Theory course at the University of Brighton, UK. Since there can be infinitely many diagrams of each knot (considering that different planes of projection can be used, and one could move parts of the knot prior to projection), identifying knot equivalence by comparing diagrams and trying to demonstrate isotopies can be extremely difficult.

Consider the educational context of a knot theory course, with a professor teaching, setting and marking assessments of students, whilst the students are learning, studying, and completing the assignments. The act of drawing of knots can be time consuming and error prone. For example, consider students trying to reproduce the knot shown in Figure 2 in class by hand; this is actually a diagram of the unknot (i.e. it is

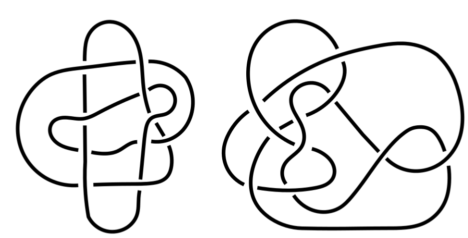


Figure 1: Are these two knots equivalent (isotopic)?

DOI reference number: 10.18293/DMS2016-035

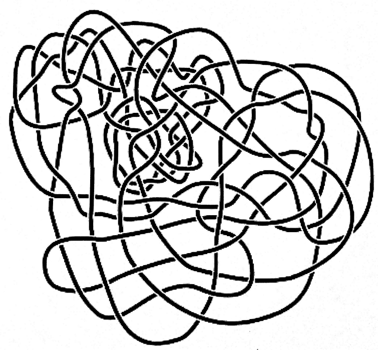


Figure 2: A knot drawing with potential for reproduction errors

equivalent to a knot diagram with no crossings, which can be depicted by a single circle), called the Haken unknot. This motivates the need for a knot drawing tool, with export facilities, and in particular the incorporation of sketch based input facilities to mimic the normal knot drawing method when constructing knots by hand.

Knot codes (like a Gauss code, discussed in Section 2.1) provide a string based representation of the crossings (and their relative orientations) in a knot diagram. In the process of understanding their usage, a student may be asked to produce the code of a given diagram or to try to construct a diagram, given a code, for relatively simple cases. The question "which signed Gauss codes are realisable as (classical) knots?" [17] was open for a long time, but algorithmic solutions have been devised (e.g. [9, 23, 28]). Kauffman [20] extended the class of classical knots (the knots we have described are termed classical knots) to virtual knots, permitting extra crossings without any over/under information, so that every Gauss code is then realisable as a virtual knot. In this paper, we focus on the classical knot case, as would be common in the majority of knot theory courses, but we adopt an underlying approach that can be naturally extended to permit the consideration of virtual knots. The related requirements for a tool are: the automatic reading of the code from a knot drawing, and the automatic generation of a knot diagram from a code.

The codes considered only determine the drawing of a knot diagram drawn on a sphere instead of on a plane; a choice of face (in the shadow of the knot on the sphere, which is the underlying graph given by forgetting the over/under information at the crossings) for stereographic projection is required to provide a drawing in the plane. However, this property can be turned into an interesting benefit. Different choices of (outer) face yield isotopic diagrams (they are, after all just different views of the same knot) but they may look very different. Thus, the alternative choices of outer face give rise to a set of diagrams which are all equivalent but may not look like it, thereby: (i) giving rise to easily constructed examples of equivalence for students to explore; and (ii) helping students to develop their understanding of stereographic projection. Without any tool support, students may find taking a diagram in the plane, considering it as drawn on the sphere and re-projecting using a different outer face, a rather challenging task. It also provides the teacher (or a professional mathematician if the context of intended usage was widened) with a means of examining this somewhat less-familiar relationship between knots that may involve very complex sequences of Reidemiester moves to realize as an isotopy.

We implemented the KnotSketch tool following these elicited requirements. We demonstrate some of its functionalities via a case study, providing a series of worked examples. We follow this up with a preliminary user study to evaluate the general usability of the tool. There are many technicalities to be dealt with in the formal setting of knot theory, but for accessibility to a Computer Science audience we adopted a relatively informal approach, skipping over some technical details (e.g. we presume all knots are tame to rule out pathological examples). An interested reader can refer to graduate level text books on knot theory (e.g. [15, 18]) for more complete details. Discussions about knots (single loops embedded in space) extend to links (disjoint unions of knots) and KnotSketch also supports links.

2. Preliminaries

We describe some facts about knots and their diagrams (informally indicating the content of well known definitions and theorems), and introduce Gauss codes, with examples, providing rationale for the chosen form adopted. A link is a disjoint union of knots, and a link *diagram* is the image of a regular projection (i.e. the only singularities are transverse double points) of the link L with over/under information added at each of the double points, called *crossings*. Every (tame) link has a diagram. Ambient isotopy is an equivalence relation on knots (or links). Each equivalence class of knots is called a *knot type*; equivalent knots have the same knot type. As is common, we abuse language and use the term 'knot' to mean the whole equivalence class (a knot type) or a particular representative. When we say that two knots are different (not equal) we mean that they are inequivalent (i.e they have different knot types). If a knot has the same type as the trivial knot then we say it is *unknotted*. We can orient a knot by nominating one of the two directions along it. If K is an oriented knot then the knot with the opposite orientation, denoted by $-K$, is called the *reverse* of K. The knot $-K^*$ is called the *inverse* of K, where K^* is the mirror image of K (obtainable by switching all crossings in a diagram of K). For oriented links, we can assign to each crossing c of a diagram, its *sign*, which is a value in ±1, denoted $sign(c)$, as depicted in Figure 3.

Let R_1, R_2 and R_3 denote the diagrammatic moves shown in Figure 4. Two diagrams differ by one of these moves if they are identical outside a small region, and inside the region they differ exactly as shown in the moves. These moves are called the *Reidemeister moves* [25]. We can think of R_3 as moving one of the strands across the crossings of the other two strands. The moves are presumed to preserve orientation. Note that we

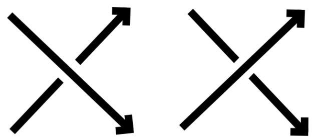


Figure 3: The overpass is the unbroken line for each crossing (on left figure: top left to bottom right; on right figure: bottom left to top right) and the underpass is the broken line. The sign of a crossing is $+$ if traversing along the the underpass, following the orientation, at the crossing the overpass is passing from the left to the right (as in the figure on the left), whilst the sign is $-$ if the overpass is passing from the right to the left (as in the figure on the right).

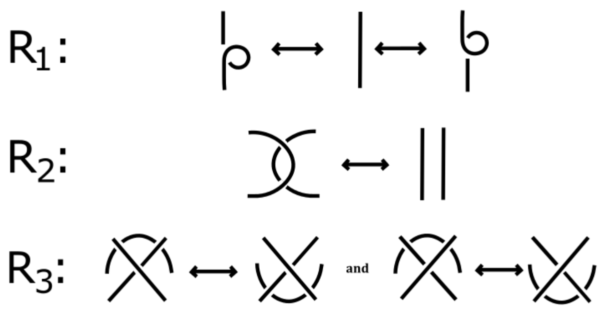


Figure 4: The Reidemeister moves, which are diagrammatic moves encapsulating knot equivalence.

interpret diagrams that differ by homotopy preserving the arc and crossing structure (i.e. the arcs can be moved without changing the underlying crossing structure; e.g. making the knot more "wiggly", or scaling the diagram) as the same.

We say that diagrams D and D' are *isotopic* if D can be obtained from D' by a sequence of moves of type R_i, with $1 \leq i \leq 3$. The diagrams are *regularly isotopic* if R_1 is not used. The following important theorem (see [25]) allows us to study knots and links combinatorially, via diagrams: Suppose that links L_0 and L_1 have diagrams D_0 and D_1, respectively. Then L_0 and L_1 are ambient isotopic if and only if D_0 and D_1 are isotopic.

2.1. Codes

Gauss codes are a means of capturing information in a knot diagram. We adopt a richer form of the codes used in the literature from which others can be recovered. The code of a diagram is given by numbering the crossings, picking an orientation on the curves and then traversing the curves one at a time, writing down the crossings met in order in a complete circuit, noting whether one was on the overpass or underpass (o/u) at each crossing along with an associated sign ($\pm$). It is well defined up to the equivalence relation generated by these choices. Codes for different components of a link are separated by a $/$.

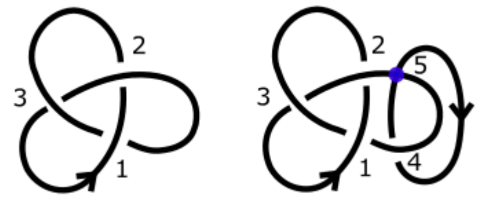


Figure 5: Left: The enhanced Gauss code $o_{1+}\, u_{2-}\, o_{3+}\, u_{1-}\, o_{2+}\, u_{3-}$ indicates the crossings met upon traversal of the knot, together with extra information (over/under, and the sign of the associated immersed curve), realisable as a (classical) knot diagram. Note that the "usual" Gauss code would assign $-$ to every crossing number, yielding: $o_{1-}\, u_{2-}\, o_{3-}\, u_{1-}\, o_{2-}\, u_{3-}$. Right: the code enables the explicit indication of the position of virtual crossings: $o_{1+}\, u_{2-}\, o_{3+}\, u_{1-}\, o_{4-}\, v_{5+}\, o_{2+}\, u_{3-}\, /\, u_{4+}\, v_{5-}$ for the 2-component virtual link. Here, we emphasized the virtual crossing by placing a dot at the crossing (numbered 5).

We describe variations of the code used in the literature. Kauffman, in [20], uses o/u, and the sign of the crossing ($+/-$ described earlier) is attached to each symbol (denoting the number assigned to the crossing), with the same sign occurring for both of the occurrences of this symbol in the code. However, Kurlin [21] includes the same sign of the crossing ($+/-$) but only attaches it to the symbol associated to the undercrossing, thereby removing the need to explicitly indicate o/u's, since the presence of a sign indicates an undercrossing, whilst its absence indicates an overcrossing. Carter [11], considers immersed curves (so the arcs pass through each other at the crossing instead of passing over and under), so there are no o/u's to consider. A variation of the sign convention is adopted, with one $+$ and one $-$ associated to each occurrence of a symbol for a crossing. An intuitive method for calculating this (enhanced) sign convention used is to imagine that the arc under consideration (as we traverse the curve and it passes through a crossing we write down one instance of the symbol for that crossing and decide on its sign) is an undercrossing and calculate the sign as per the earlier method for knots. This leads to one of the two symbols being assigned to a crossing being a $+$ and the other a $-$.

We make use of a variation, which we call the *enhanced Gauss code* (*code* for short) in which we use the sign convention for immersed curves, following Carter, as well as the o/u information (as used by Kauffman) for knots. This generalisation provides us with the ability to deal with curves that do not have any over/under information at the same time as those that do. One application is to explicitly encode virtual crossings in virtual knot diagrams (rather than simply ignoring their presence in the code), permitting the expression of over/under or "through/virtual" at each crossing. The "through/virtual" option can naturally be represented by a v for virtual crossing in the virtual knot diagram setting. Whilst we do not explicitly consider the use of virtual knots within the educational con-

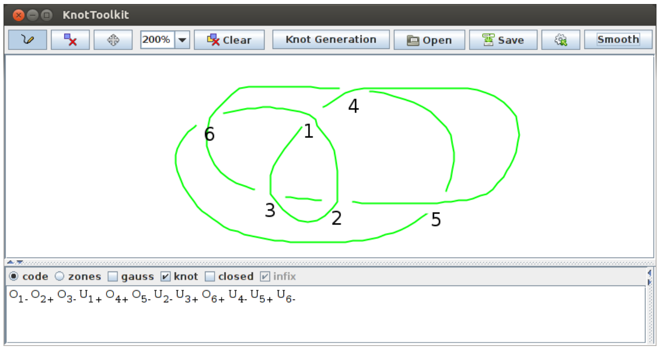


Figure 6: A screenshot of the KnotSketch interface, with *toolbox* at the top, a diagram shown in the *Sketch View* and its automatically produced code in the *Code View*.

text in this paper, we envisage the future usage of this functionality (for advanced postgraduate students or professional mathematicians), and so we decided to adopt a notational convention that would be consistent with this.

Figure 5 shows examples of the (enhanced Gauss) codes of a classical knot diagram (top) and a virtual link diagram (bottom). To enable a comparison with the immersed curve codes, one can consider the shadow of these classical knot diagrams (i.e. make all crossings "through/virtual"); the effect of this on the code is to simply remove the occurrences of o/u, replacing them with v's. This would have necessarily been slightly more complicated if we had used the classical knot sign convention in the code. KnotSketch enables the automatic interpretation of the code of (sketched) link diagrams.

3. KnotSketch

We describe KnotSketch and its main functionalities. KnotSketch is partly based on ink recognition techniques, previously developed for other applications [6, 13, 14].

3.1. User Interface

As shown in Figure 6, the KnotSketch interface is divided into three parts. The upper part is a *toolbox* with buttons to perform the following operations (from left to right):

- change the input mode to *draw*;
- change the input mode to *erase*;
- change the input mode to *move*. At present it is only possible to move whole curves, not parts of a curve (but this would be a useful future additional functionality for interaction);
- change the zoom level;
- clear the current drawing;
- launch the automatic diagram generation facility through a new dialog. The dialog (see Figure 7) contains a text field to enter the code. It is possible to use the clipboard to paste the automatically produced code of a sketched diagram. The code specifies the drawing of the knot on a sphere. By clicking on the OK button, a new dialog is opened (an example is presented in Figure 10 of Section 4). This dialog requires the user to choose a face assigned to be the outside region under stereographic projection from the sphere onto the plane from a list of possible choices (one for each region). A preview pane is shown by the side of the list to assist the user. Moreover, it is possible to personalize the diagram appearance with different visualization options, such as varying the number of spring embedding iterations;

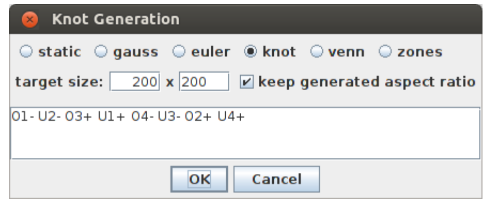


Figure 7: Code to diagram generation box.

- open a previously saved diagram. If a file describing a diagram drawing (see below) is opened then its content is shown in the *Sketch View* described below, while if a file containing the knot code is opened the diagram generation is launched;
- save the current diagram in a native drawing file format. It is also possible to export a drawing in *svg* format (optionally as an *html* file), whilst the code can be exported in *txt* or *html* format. This facilitates the use of the tool for the production of diagrams that occur in research papers, student submissions, or to include in web-pages;
- open the Options window in order to alter tool settings, such as the size of the gap used in the visualisation of the crossings;
- perform knot beautification using EulerSmooth [30]. Given a drawn diagram, the user can apply a smoothing operation via EulerSmooth to improve the quality of the sketched drawing; EulerSmooth was defined to work with Euler diagrams but it can be heuristically applied to knot diagrams. The smoothing function can be manually controlled and can be applied with various parameter settings.

In the middle part of the KnotSketch interface the *Sketch View* contains the drawing canvas. The curves are arbitrarily shaped and must be completed with a single pen stroke. Once a stroke has been entered, its endpoints are automatically joined to close the curve. If the user tries to make the curve closed by passing the final part of the curve over the initial part, then any additional crossing created is erased. Upon its completion, a curve is coloured with a colour chosen from a pre-defined list (the predefined colour list can be configured by the user). As soon as a curve is completed, a numeric label is automatically assigned to each crossing point. There are two segments passing through each crossing, and by default the last drawn segment is shown as an over-crossing (as a future option, the user will be able to choose the default behaviour, permitting the automatic provision of alternating crossings as the knot is traversed, for example); the crossing type (over/under) can be manually changed by the user by clicking on the crossings. The length of the gap used to indicate the crossing can also be configured by the user.

In the lower part of the interface, the *Code View* contains the automatically generated enhanced Gauss code, as described in Section 2.1 (it is possible to select the code and copy it to the clipboard). At the top of this view a toolbox provides a set of options through which the user can select some other forms of codes (that are not required for this particular application).

A user study is discussed in Section 5, and Figures 8-12 show some related screenshots of the tool.

3.2. Back-End

KnotSketch is written as a Java 7 application. Other than knot drawing, the main features of the application are knot interpretation (i.e. computing the code of a drawn knot), knot generation from code and knot beautification. The enhanced Gauss code, displayed in the *Code View*, is incrementally constructed and is updated every time a curve is added or deleted. It is stored in an internal format enabling efficient operation executions. A knot is represented as a closed polyline.

KnotSketch can generate knot drawings from an enhanced Gauss code using a planar graph based construction: the graph contains one node for each crossing, and its combinatorial embedding of the graph can be calculated by using the crossing signs $(+/-)$ and the choice of face assigned to be the outer region. Then an algorithm from the OGDF framework [24] can be used to embed the planar graph in the plane. The user can optionally modify the embedding by selecting a number of iterations of a force directed algorithm [5] to apply. The algorithm theoretically preserves the regions, but approximation due to rounding could cause non-preservation. However, the tool provides a post-check if the initial and final diagram codes are equivalent. Finally, by traversing the graph edges appropriately, the curve for the knot diagram is constructed.

The knot beatification step can be performed by using the Ocotillo java library from EulerSmooth. The drawn knot is treated as if it was an Euler diagram (essentially taking the shadow of the knot, viewing all crossings as if they passed through instead of over/under each other; note that the constraints of non-self intersections for Euler diagrams must be relaxed in this context). The knot diagram is converted to the format used by Ocotillo and the beautification process is performed on it, using the parameters set by the user. After each beautification iteration the diagram is converted back to the original format in order to display the beautification progress to the user. Future enhancements of EulerSmooth in this context should lead to enhancements of the outputs.

4. Case study

We describe a set of activities that can be performed with KnotSketch within an educational context, demonstrating the capabilities of the approach and the tool. We highlight a novel viewpoint that permits an uncommon form of user exploration, which would be challenging for students to perform without tool-based assistance. We discuss general activities that users may undertake along with low level tasks that users may perform along the way.

Consider the task of starting with a code and creating a diagram of a knot with that code. This task can be used to test a users' understanding of the code to diagram process, and to highlight how challenging the construction can be without further assistance. In a simple case, the user can directly sketch the requested diagram. They can check their solution via the automatic production of the code, displayable in the *Code View*. If the knot produced differs from the required solution via crossing changes alone (i.e. switching over and under crossings) then user interaction via clicking on the crossings can be applied to alter the diagram accordingly. If the diagram drawn has the wrong code and the code differs by more than crossing changes then the curve needs to be redrawn. An alternative approach is to use the automatic diagram generation facilities from a code (described in Section 3). For the task of producing a diagram with the correct code, any choice of outer face will do, so the user can select any of the proposed faces. This functionality leads to an interesting possibility for the exploration of knot equivalence (isotopy). Given a code one can generate different views of the knot (one for each face, by using the different choices of outer face in the plane of projection). Since each of the views are diagrams of the same knot viewed from a different perspective, all of these diagrams are equivalent. However, they may look very different and it may be challenging for a user to construct an isotopy (a sequence of Reidemeister moves) from one diagram to the other. Thus, given two knot diagrams that potentially look very different, but which are in fact different projections of the same knot, the user can try to identify this knot equivalence by searching through the options and visually comparing the outputs.

Figure 8-12 show diagrams from the tool similar to those of the case study. In detail, Figure 8 shows a user sketched (and smoothed) drawing of a trefoil knot with 3 crossings, its automatically produced code, along with a more complicated knot with 7 crossings. Figure 9 shows diagrams of a knot K, the knot K' with a single crossing change, and, $m(K)$, the mirror of the knot K, which is shown as K with all crossings changed; the effects on the code are also shown. Figures 7 and 10 show an example of a code being used to generate a knot, demonstrating the user's view at the time of selection of outer face. The code-like information displayed for each choice is a means of specifying the region (details are omitted since they are unimportant for our purposes here); we display drawings arising from all six choices of outer face for this code. Figure 11 shows diagrams (far left and far right) and their (not obvious) equivalence as a sequence of Reidemeister moves (indicated by R_i's). These two diagrams can be seen to be equivalent via projection onto different faces; the left and right diagrams are those shown in Figure 10(f) and Figure 10(e), respectively, up to moving the arcs without changing

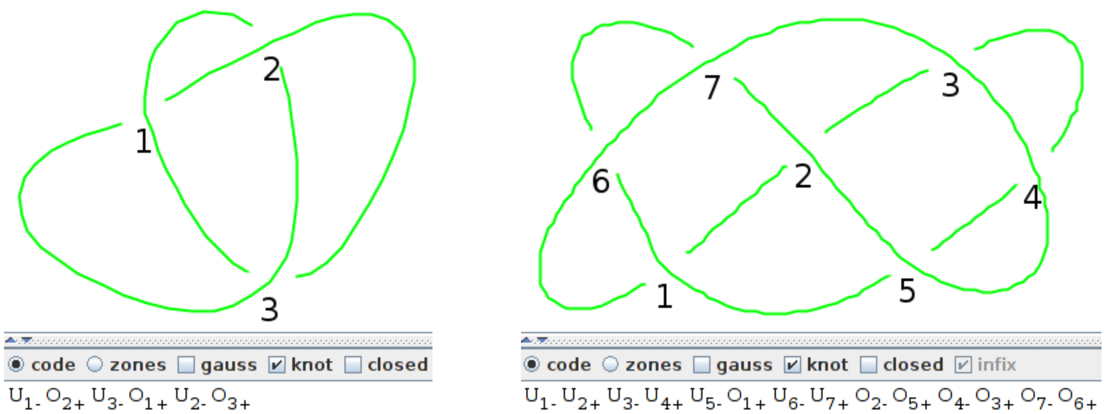


Figure 8: Screenshots of a trefoil knot (left) and a knot with seven crossings (right) and their codes.

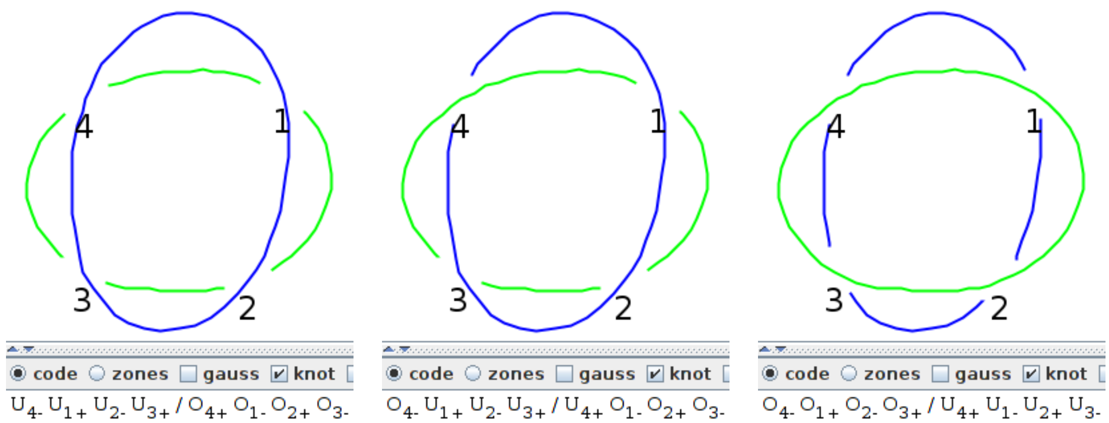


Figure 9: A diagram of a link (left), the diagram obtained by changing crossing 4 (middle), and the mirror of original diagram, given by changing all crossings (right).

the crossing structure. Figure 12 shows an example of generation of a knot diagram from a complicated code, showing the initial creation of a knot, along with an initial improvement using the EulerSmooth functionalities.

5. Experimental setup

We performed a preliminary evaluation on a student population to test the usability of the tool itself. In particular, we measured the perceived usability of the system through a System Usability Scale (SUS) [7] questionnaire. The questionnaire is composed of 10 statements to which participants assign a score indicating their strength of agreement on a 5-point scale. The final SUS score ranges from 0 to 100. Higher scores indicate better perceived usability. We also gathered some participants' free-form comments during and after the experiment.

The demographic mostly consisted of Computer Science students, which may help with familiarity with the use of software. In the future, we will also examine the usage with Mathematics students whom may be more familiar with the mathematical context, but perhaps less proficient in software usage. We recruited a total of 10 participants (3 female). The ages ranged from 21 to 51 (with mean $M = 31$, and standard deviation $SD = 10.4$). All of them were habitual computer users and had previous experience with touch-screens. None of them had any prior knowledge of knot theory.

The set of tasks we considered were as follows.

TASK 1 Given the following knot code, generate a diagram of it using the "generation from knot code" functionality of KnotSketch: $o_{1+}\ o_{2-}\ u_{3+}\ u_{1-}\ u_{4+}\ u_{5-}\ o_{5+}\ o_{4-}\ u_{2+}\ o_{3-}$;

TASK 2 Given the knot drawn on the sheet, reproduce it using KnotSketch and save its knot code in a text file. Moreover, save the drawing in the tool's native file format (this task was repeated three times - TASK 2.1, TASK 2.2, TASK 2.3 - once for each knot in Figure 13);

TASK 3 Open the native format of the drawing previously produced in TASK 2.2 and generate its projection onto a

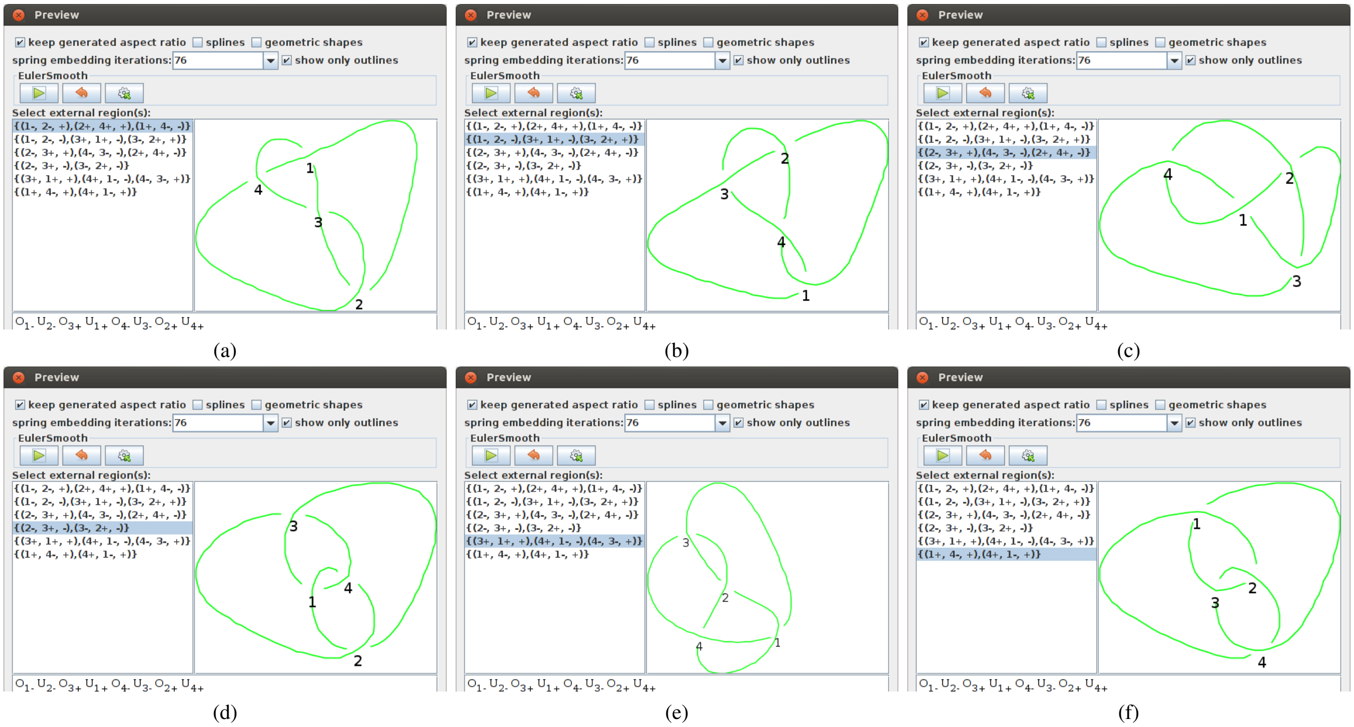


Figure 10: Displaying the preview of the different choices of outer face for the projection, using the code from Figure 7.

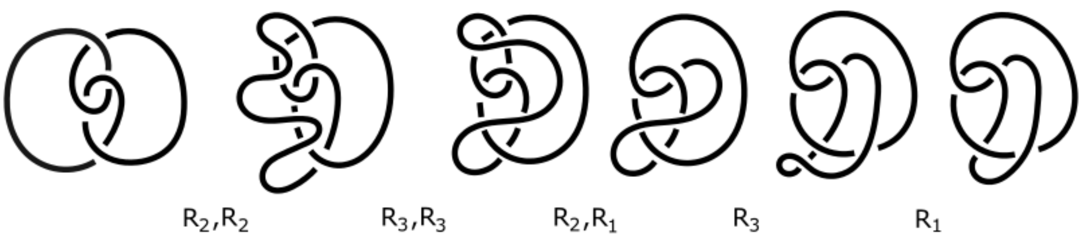


Figure 11: A sequence of Reidemeister moves demonstrating the equivalence between the left diagram (c.f. the diagram in Figure 10(f)) and the right diagram (c.f. the diagram in Figure 10(e)).

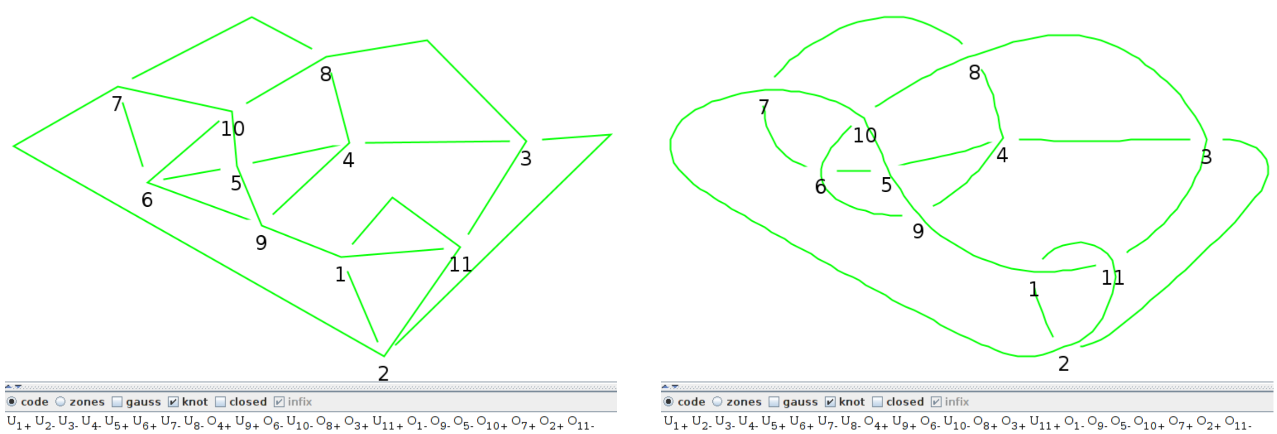


Figure 12: Generation of a knot diagram from a complicated code, with the initial creation of a knot (left), along with an initial improvement using the Eulersmooth functionalities (right).

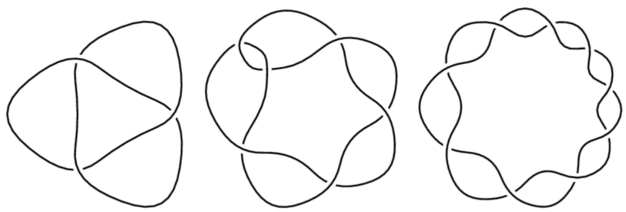
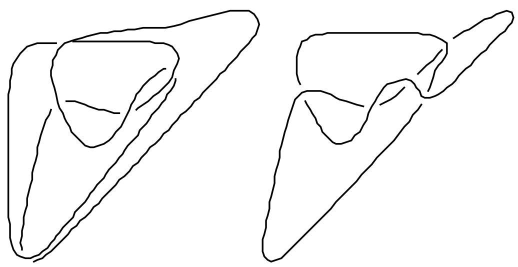
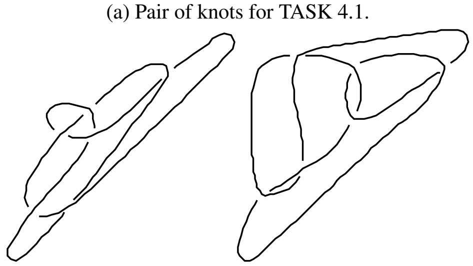


Figure 13: TASK 2 knots.



(a) Pair of knots for TASK 4.1.



(b) Pair of knots for TASK 4.2.

Figure 14: TASK 4 knot pairs.

Task	Avg time (sec.)	S.D.	User errors
TASK 1	64.7	16.6	0
TASK 2.1	45.3	14.1	0
TASK 2.2	60.2	13.3	0
TASK 2.3	137.6	72.8	2
TASK 3	20.0	11.8	0
TASK 4.1	135.0	48.8	2
TASK 4.2	152.6	41.6	0

Table 1: Task completion times.

Username	Score
participant 1	77.5
participant 2	85
participant 3	62.5
participant 4	75
participant 5	87.5
participant 6	75
participant 7	85
participant 8	77.5
participant 9	92.5
participant 10	85
Average	80.25

Table 2: SUS-like questionnaire scores for user satisfaction.

different outer face;

TASK 4 Given the pairs of knot diagrams on the sheet, indicate if they differ by redrawing via projection onto a different outer face or not (the task was repeated twice - TASK 4.1, TASK 4.2 - once for each knot pair in Figure 14).

The time limit for the completion of each subtask was 5 minutes.

The tasks were executed on a *Dell Precision T5400* workstation equipped with an *Intel Xeon* CPU at 2.50 GHz running *Microsoft Windows* operating system and the Java Run-Time Environment 8. The device used for the experiment was a Sympodium ID250 Interactive Pen Display, attached through both USB and RGB cables to the work station.

5.1. Results

The tasks have been successfully completed by almost all participants. Only two tasks were not completed successfully by all participants: TASK 2.3 and TASK 4.1, with two errors each. The mean task completion times are reported, along with standard deviations, in Table 1. As expected, the subtasks of TASK 2 showed increasing difficulties; in particular for TASK 2.3, some users needed to attempt the drawing several times. However, the error on this task was simply an incorrect sign on a single crossing. TASK 4 presented some difficulties for most participants, with relatively high execution times and two errors for TASK 4.1.

The scores of the questionnaire calculated from the responses of the participants are shown in Table 2, indicating scores ranging from 75 to 92.5, with an average value of 80.25. This value indicates a good level of satisfaction [4]. An analysis of the questionnaire scores showed that the participants judged the tool very useful for its purpose, but they did not find all the features simple to use and some participants needed some learning time to be able to master them. For instance, some participant lamented the lack of feedback on the presence of errors in a written code; others had problems in finding out how to generate the projection of a drawing onto a different outer face.

Similar considerations emerged after a brief interview/discussion with the participants involved in the experiment to understand what aspects of the tool were of help in the execution of the tasks: probably due to unfamiliarity with Knot Theory, some participants expressed initial doubts about the comprehension of the tool's purpose. Furthermore, some of them had difficulty in completing TASK 4 even with the help of the tool, with one participant stating: "I would have preferred more support from the tool to perform TASK 4". Nevertheless, he declared "I would never be able to complete that task without the help of the tool".

Tool	KnotSketch	KnotPlot [26]	Knot ID [2]
Hand sketch	Yes	Yes (but requires license)	No (point and click/drag)
3D visualisation	No	Yes	Yes
Link diagrams	Yes	Yes	No
Save/Load/Export diagram	• Supports saving and loading diagrams • Diagrams can be exported in SVG and HTML format	• Supports saving and loading diagrams • Diagrams can be exported in EPS format	• Load list of coordinates • Load gauss code (no visualisation of the knot in this case)
Edit diagram	• Supports adding, deleting and moving individual components	• Supports adding, deleting and moving individual components or a segment of any component	• Supports adding, deleting or modifying individual segments of any component
Diagram generation via code	Supports Gauss Code	Supports Dowker code and Braid word	No generation (but identification available from Gauss Code)
Diagram reprojection via outer face	Yes	No (but rotation of 3D visualisation is possible)	No
Diagram smoothing	Yes	Yes	No

Table 3: Visualisation features of KnotSketch, KnotPlot and KnotID

6. Related Work

Existing knot tools have varied functionalities. Table 3 provides comparison of visualisation aspects of KnotSketch with KnotPlot [26] and Knot ID [2]. We provide some further details and briefly discuss other notable tools below. To the best of our knowledge, the re-projection facility onto different outer faces offered by KnotSketch is novel, as is the capability to deal with (sketching, interpreting) virtual knots or links (note that the study was restricted to classical knots and links).

KnotPlot [26] is a widely used program for visualising and interacting with knots that has theoretical underpinnings as described in [29]. A knot can be sketched by hand (in the upgraded version requiring a license) or constructed from code using a tangle notation system developed by Conway [12], or from a braid word, or a Dowker-Thistlethwaite code (which does not uniquely specify composite knots [26, page 98]). A knot can be refined into a smoother configuration using a 3D technique, which is computationally expensive, especially for large knot diagrams with more than 50 crossings [26, page 105]. KnotPlot has a built-in database containing a wide range of knots and links which can be viewed, modified and saved, along with other features such as: computing the Alexander and HOMFLY polynomials, the writhe, the average crossing number, the thickness and the Dowker code of a knot; searching for minimal stick conformations and interesting random knots; generation of arbitrary braids; enabling the consideration of open knots or links; interactive construction of knotted and linked spheres in four dimensions.

KnotID [2] is a web application that allows the viewing of topological information about knots. The application can be used: (1) to import three-dimensional curves (as a list of 3D coordinates or by generating torus knots from two coprime numbers); (2) to draw two-dimensional curves; (3) to directly input an enhanced version of the Gauss code similar to those described in Section 2.1. In addition to displaying topological information, the tool can, through the application of topological invariants, identify if the input knot is equivalent to knots found in a lockup table (based on [1]). It permits computation of properties such as the Reduced crossing number, Determinant $|\Delta(-1)|$, $|\Delta(\exp(2\pi i/3)|$, $|\Delta(i)|$, and Vassilev invariants. Compared to KnotSketch, the software offers a non-sketching oriented drawing interface and no features such as delete/smoothing/load/save or the ability to display the crossing numbers. Moreover it does not offer functions to generate a knot from a gauss code or to regenerate a knot by re-projection onto a different plane.

Libbiarc [10] can be used to manipulate and analyze properties of knotted curves, compiled into a $C++$ library; the most frequently used functions are available as an API. The core library provides functions to interpolate point-tangent data with bi-arcs, access information such as curvature and torsion, compute the length and thickness. The library includes a viewer application, called curview, for visualisation, in which a knot can be loaded, manipulated and saved.

KnotApp [31] is a thesis that describes a program that displays a knot and other objects, such as the knot's crossing map and its trisecant curve, using the jReality JOGL Viewer. The knot may be chosen from a provided list or loaded from the file system. Then the chosen knot can be edited by dragging its vertices using the provided knot editor. The crossing map implemented displays the set of irregular projections of the underlying knot as curves on the unit sphere. The set of trisecants of the underlying knot is visualised as a curve in the 3-dimensional torus. However, the KnotApp application itself does not appear to be currently available to be tested.

Other notable knot tools, with varying functionalities include Linknot (see [19]), Knotscape [32] (uses Dowker-Thistlethwaite codes), and MING [33] (which minimizes MD-energy of polygonal knots, reduces the numbers of edges, and draws/visualizes their 3-dimensional pictures).

7. Conclusions

We have developed KnotSketch, a tool to facilitate knot exploration, with particularly interesting functionalities of permitting knot sketching followed by classical knot diagram regeneration via re-projection onto different planes via stereographic projection from the diagram drawn on the sphere. This could be particularly useful in University level educational settings, and we examine the potential capabilities via a case study. In addition, we adopt an enhanced version of Gauss code that enables the future handling of virtual knots. This ability to deal with virtual knots would likely be useful in the context of expert users, such as professional mathematicians making use of the tool in exploratory mathematics or simply to produce figures for use in publications (the use of classical knots is of course pertinent to this user class as well). One can also envisage different semantical interpretations of these types of diagrams with mixed crossing types, and this would widen the applicability even further beyond the consideration of knots.

We performed a preliminary user-based empirical study to gain insight into the usability of the toolkit, with an aim for future improvements. We plan to perform more extensive user testing, and adopt an iterative developmental methodology taking into account user-insights after testing periods. We wish to explore the utility for both Mathematics students and professional Mathematicians. Using the interface at KnotInfo [3], a user can select sets of knots and request information about them including their diagrams or compute many invariants. This interface and database (and others) could be made to be usable in conjunction with KnotSketch, thereby greatly enhancing its functionalities.

References

[1] Knot atlas take home database. At `http://katlas.org/wiki/The_Take_Home_Database`.

[2] Knot ID website. At `http://inclem.net/knotidentifier/`.

[3] Knot info website. At `http://www.indiana.edu/~knotinfo/`.

[4] A. Bangor, P. T. Kortum, and J. T. Miller. An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction*, 24(6):574–594, 2008.

[5] F. Bertault. A force-directed algorithm that preserves edge-crossing properties. In *Graph Drawing*, volume 1731 of *Lecture Notes in Computer Science*, pages 351–358, 1999.

[6] P. Bottoni, G. Costagliola, M. De Rosa, A. Fish, and V. Fuccella. Euler diagram codes: Interpretation and generation. In *Proceedings of the 6th International Symposium on Visual Information Communication and Interaction*, VINCI '13, pages 105–106, New York, NY, USA, 2013. ACM.

[7] J. Brooke. Sus: A quick and dirty usability scale. In P. W. Jordan, B. Weerdmeester, A. Thomas, and I. L. Mclelland, editors, *Usability evaluation in industry*. Taylor and Francis, London, 1996.

[8] D. D. Buck. EP/H031367/1 SANDPIT : Knots and Evolution - Topologically Driven Integrase Mutagenesis, and EP/G039585/1 DNA Knotting and Linking: Applications of 3-Manifold Topology to DNA-Protein Interactions, 2009.

[9] G. Cairns and D. Elton. The Planarity Problem II. *Journal of Knot Theory and its Ramifications*, 5:137–144, 1996.

[10] M. Carlen. *Computation and Visualisation of Ideal Knot Shapes*. Ph.d. thesis, EPFL, Lausanne, 2010.

[11] J. Carter. Classifying immersed curves. In *Proc. Amer. Math. Soc*, volume 111, pages 281–287, 1991.

[12] J. H. Conway. An enumeration of knots and links, and some of their algebraic properties. *Computational Problems in Abstract Algebra*, pages 329–358, 1970.

[13] G. Costagliola, M. De Rosa, A. Fish, V. Fuccella, and R. Saleh. Curve-based diagram specification and construction. In *Proc. VL/HCC*, pages 39–42, 2013.

[14] G. Costagliola, V. Fuccella, and M. Di Capua. Interpretation of strokes in radial menus: The case of the keyscretch text entry method. *Journal of Visual Languages & Computing*, 24(4):234 – 247, 2013.

[15] P. Cromwell. *Knots and Links*. Cambridge University Press, 2004.

[16] M. Dennis and A. Taylor. Vortex knots in tangled quantum eigenfunctions. *Nature Communications*, (12346), 2016.

[17] C. Gauss. Werke. *Band 8. Teubner*, 1900.

[18] N. Gilbert and T. Porter. *Knots and Surfaces*. Oxford Science Publications, 1997.

[19] S. Jablan and R. Sazdanovi. *LinKnot: Knot Theory by Computer*, volume 21 of *Series on Knots and Everything*. World Scientific, 2007.

[20] L. Kauffman. Virtual knot theory. *European Journal of Combinatorics*, 20:663–691, 1999.

[21] Kurlin. Gauss paragraphs of classical links and a characterization of virtual link groups. *Math. Proc. Camb. Phil. Soc.*, 145:129–140, 2008.

[22] R. Metzler. Knots, bubbles, unwinding, and breathing: Probing the topology of DNA and other biomolecules. *Handbook of Theoretical and Computational Nanotechnology, edited by M. Rieth and W. Schommers*, 1:1–54, 2005.

[23] J. Nagy. Über ein topologisches Problem von Gauss. *Mathematische Zeitschrift*, 26(1):579–592, 1927.

[24] OGDF. Open graph drawing framework. `http://www.ogdf.net`.

[25] K. Reidemeister. Elementare begrøundung der knotentheorie. *Abh. Math. Sem. Univ. Hamburg*, 5:24–32, 1926.

[26] R.G.Scharein. *Interactive topological drawing*. Ph.d. thesis, The University of British Columbia, 1998.

[27] D. Rolfsen. *Knots and Links.* AMS Chelsea Press, 2003.

[28] P. Rosenstiehl and R. Tarjan. Gauss codes, planar Hamiltonian graphs, and stack-sortable permutations. *Journal of algorithms*, 5(3):375–390, 1984.

[29] R. Scharein. *Interactive Topological Drawing*. PhD thesis, University of British Columbia, 1998.

[30] P. Simonetto, D. Archambault, and C. Scheidegger. A simple approach for boundary improvement of Euler diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):678–687, 2016.

[31] M. Sommer. *Visualization in Geometric Knot Theory - Understanding the mathematical structure of trisecants*. Ph.d. thesis, Technische Universität Berlin, 2007.

[32] M. Thistlethwaite. Knotscape. At `http://www.math.utk.edu/~morwen/`.

[33] Y.-Q. Wu. Ming. At `http://www.math.uiowa.edu/~wu/`.

Visualizing student engagement in e-learning environment

Enrica Pesare, Teresa Roselli, Veronica Rossano
Dipartimento di Informatica
Università degli Studi di Bari Aldo Moro
Bari, Italy
{enrica.pesare, teresa.roselli. veronica.rossano}@uniba.it

***Abstract*— The learning assessment in e-learning contexts is one of the latest challenges for educational technology researchers. One of the main issues to be addressed is the definition of dimensions that should be used to measure the learning effectiveness. In this perspective, the research work aims at defining the engagement indicators useful to assess the active participation of students in social learning environments. Moreover, the paper presents the design and implementation of Learning Dashboards aimed at visualizing the student engagement in online communities where the engagement and involvement of students are the key factors for successful learning.**

Keywords- Assessment; Learning Analytics; Learning dashboard; Social learning environments; Engagement.

I. INTRODUCTION

The learning assessment is one of the main issues in the educational field. In the latest years, the interest in this topic is rising thanks to the evolution of e-learning methods and techniques. The new teaching models adopted in the MOOCs (Massive Open Online Courses) [1] [2] require the adoption of new tools and measures that allow teachers to make effective and reliable assessment even with a large population of learners. In the literature, different solutions have been proposed, such as the adoption of qualitative measures [3] [4] to have detailed information about the interaction of the learner with the peers and the didactic resources, or the definition of tools able to supply detailed information on the student learning path, starting from the tracking data of e-learning platform [5] [6]. For these reasons, the interest of the educational technology researchers is focusing on Learning Dashboards that allow, both faculty and students, to tune their actions (lectures, assignments, study) based on rapid feedbacks on student progress in learning [7]. Currently, the interactions are becoming even more complex in the e-learning environments. Thus, the number of dimensions to take into account to make a successful assessment of the student's learning path is improving. The main challenge of Educational Technology field is to point out all the dimensions useful to measure the effectiveness of the e-learning paths [8].

DOI reference number: 10.18293/DMS2016-028

In this perspective, our point of view is that in the e-learning environments 2.0, where the construction of the knowledge is shared among the different users, to measure the engagement in the e-learning process could give information about the learning effectiveness. For this reason, the paper presents a Learning Dashboard aimed at visualizing the student engagement in web based learning environment. In particular, the attention is focused on online communities, where the engagement and involvement of students is the key factor for a successful learning.

II. RELATED WORKS

The Learning Dashboards are commonly used in a wide range of e-learning environments: Learning Management Systems, Web-based and Personal Learning Environments, Massive Open Online Courses and so on. Information Visualization techniques, indeed, are powerful tools in the learning analytics research field, since they allow to visualize the collected data about student's activities. Visualization can impact on the user behavior and motivation, for both students and teachers, and promote self-awareness and reflection about the learning process [9].

The interest in this research field and the number of proposed approaches are growing in the latest years. For ours research goals, we decided to narrow the research on two important aspects in e-learning environments: student's progresses and performances and student's engagement.

A. *Visualizing progresses and performaces in e-learning enviroment*

Numerous are the solutions proposed to face this issue. Student Activity Meter (SAM) [10] [11], for example, was designed to explore the classroom activities. It provides an overview of student's activities through simple statistics. For each student, the indicators are the time spent and the documents used. These indicators are compared with minimum, maximum and average values of the whole class.

The first indicator, time spent, is displayed with a line chart visualization; it *shows a line for every student in the course; the horizontal axis shows the dates and the vertical axis displays the total time spent* [11]. Indeed the visualization of activities over time is a key aspect when analyzing student behaviors. The second indicator, document used, is not directly displayed but it is used to provide recommendations with a simple recommender system.

Furthermore a parallel coordinates chart shows correlations among *(i) the total time spent on the course, (ii) the average time spent on a document, (iii) the number of documents used and (iv) the average time of the day that the students work* [11] allowing the discovery of patterns in the student's behavior.

Mastery Grids introduces the so called "social progress visualization" [12] to engage students and to guide them through the learning resources exploration. The indicators used are the progress made by each student. The progresses are presented in a matrix where for each domain topic (horizontal dimension) and kind of resources available for that topic (vertical dimension) the level of completeness *or level of progress a learner has for the given combination of topic and resource type* is showed through color intensity (third dimension).

In addition to the individual progress visualization, Mastery Grids enables comparison with other peers (the class average, the top ranking students) and highlights the differences between the individual user and the group. Furthermore, it enables more compact and detailed comparison selecting one kind of resource. This direct comparison pushes the user to improve her/his activity by stimulating them to complete different kind of activities and to access new content at the same time.

CAM Dashboard was designed to allow the exploration of learners' behavior in Personal Learning Environments and to enable both self-reflection and comparison with peers to improve student's motivation [13] [14]. It collects data from different data sources even outside the traditional LMS, and provides visualization according to the student's goals.

The indicators are the total number of activities done by the student and the number of events or time spent for each application used by the student over time. Furthermore those indicators are grouped by day of the week, by action performed or by resources involved. Noteworthy is the distinction of different kind of actions that enables a deeper level of analysis, for instance the distinction of active and passive actions such as writing and reading activities.

The main visualization is a line chart of the activity over time (annotated timeline) while the grouped indicators are presented with bar chart visualizations that allow the comparison between the time spent by the user and the average time of peers.

VeeU was designed to enhance student assessment in distance learning environment for both teachers and students. [5]. The indicators proposed are the number of daily accesses, the number and distribution of activities in a course and the completion rate of course activities.

The accesses are presented in a time line visualization at different aggregation levels (different courses or single course) for the teacher, while the student visualizes the number of her/his daily accesses compared to the average values for the class. A pie-chart visualization is used for the percentage and distribution of activities while the completion rate is displayed in a gauge chart visualization with a list of recommendations to help student in achieving goals.

Even StepUp! has been designed for students empowerment in open learning environments [15][16][17] in the "quantified self" perspective [18]. It collects tracking data from group blogs and twitter (post, comments, tweet). Even in this case the indicators are the number of activities and the time spent but the activities are also classified in assimilative (blogging and writing reports), communicative (twitter and comments) and productive activities (programming) [19] enabling distinction of active and passive behavior in this case too.

The indicators are presented with numerical data and sparklines for every student to provide a quick overview. The sparklines can be detailed in a stacked bar chart visualization that displays activity over the weeks, grouped by kind of activities and participation to promote awareness of what students did and how they spent their time.

TrAVIS [20] collects data about communication activities in distance learning environments to promote self-monitoring. It distinguishes four levels of interaction (aggregation, discussion, cooperation and collaboration) that correspond to four levels of indicators:

- aggregation level: connection frequency, threads started, messages posted, messages replied, and messages quoted. These indicators are commonly used to describe the activities of each individual student;
- discussion level: browsing, forums, posting, reading and chatting activities. These indicators are useful to estimate the user interest in the discussions or her/his level of community interaction;
- cooperation level: thread started, new messages, replied messages, quoted messages, files uploaded, files downloaded and participation level. These indicators are useful to evaluate the contribution of each user to the community.
- collaboration level: even these indicators are constructed from the lower levels indicator but in this case the focus is moved from the individual perspective to the group level perspective. As a matter of fact they are collected at group level within a defined timespan in order to compare the participation rates or the productivity rates of different groups.

The visualization technique at each level is the spider chart: a spider chart for each user (at the aggregation, discussion and cooperation levels) or for each group (at the collaboration level) allows visual comparison among students and among groups.

B. Visualizing engagement in e-learning enviroments

In the e-learning field, the increasing interest in the engagement dimension has led to the definition of different dashboards to monitor and improve student engagement.

VisEN, for example, is addressed to students to allow the exploration of data about engagement in form of visual narratives [21] [22]. It presents the student's engagement in a quick gauge chart while further details are provided on demand. Students can interact with visualizations to realize how engagement score was calculated, based on course interaction (page click), study time (reading and reviewing durations), submissions and questionnaire scores. They can compare their engagement with peers (all class members or similar engaging peers) at different levels (global score or activity level).

Even in the emerging field of MOOCs, student engagement has acquired a great importance. Coffrin [23] proposes the analysis of student's activities in MOOCs using visualization techniques. In this case, the visualizations are used at macro level to understand patterns of student engagement. However, even if the proposed analysis are not strictly related to our goal, they give us some interesting points of view on the topic. The analysis starts from two simple histograms of students' participation and assessment performance, broken down by week. This first visualization confirms the common pattern of high interest at the beginning with increasing attrition rate over time, but breaking down the data, a deeper understanding of engagement is provided. In particular, students in a MOOC can be classified into Auditors, Active and Qualified students according to the kind of activities performed (video lectures and assessments). This breakdown can be useful in different visualization (histogram or state transition diagram) to closely analyze the percentage and relative proportion of students and to understand the temporal evolution of engagement trajectories [24].

Apart from individual participation in MOOC, even the social interactions have a great influence in particular when involving social learning processes [25].

Schreurs [26] proposes the application of Social Learning Analytics in a MOOC platform. In particular, it focuses on the network visualization from discussion forums: every user is a node, every reply to a post is a tie, while a simple tag cloud allows the filtering of ties and nodes based on the contents. Moreover a tooltip will allow the discovery of topic of interest and expertise of every user. Then different network perspectives can promote reflection on learners' interaction or make visible the contents developed through discussions.

Another perspective on engagement in web-based learning environment is given in [27] where the authors propose the visual representation of cognitive and behavioral indicators of engagement to support teachers in monitoring learners' engagement. To reflect the multidimensionality of engagement the indicators are built from student participation actions (behavioral indicators) and from modification actions on the learning documents structures (cognitive indicators). The first ones are computed from number of actions and duration (number of login, number accesses to a learning resource, time spent on a resource etc.) while the latter are computed from a particular subset of action related to the structural modification of learning documents (create, add, update, delete, move and insert).

The proposed visualization is based on the small multiple visualization technique for each student's session: each frame represents a simplified mind map at different times (t_i) with dotted lines for deleted elements and solid lines for existing ones. The small multiple representation can be combined with a linear representation for each node where the structural modifications on the selected node are made visible over time. The small multiple visualization might have problems in case of growing numbers of nodes or growing number of time intervals while the linear representation is easily manageable for temporal data even if it does not allow comparisons of data about different nodes at the same time.

III. VEEU 2.0

The main goal of our research is to make students and teachers aware of their engagement in a social learning environment. As a matter of fact, teachers and students need to be aware of what kinds of interactions are occurring in the virtual space and how the building up knowledge process happens. This is the so called "situational awareness" that, according to Few's principle design [28], is one of the main purposes of dashboards. In this perspective, the rapid perception of information through the dashboard is fundamental to facilitate the decision-making process.

To achieve these goals, in the following sections are briefly presented the social learning environment and the dashboard design process, from the data analysis (to define the most suitable predictors and indicators), to the selection of the best visualization techniques (to depict the relevant data at a glance).

A. Social Learning Environment

The learning environment, as depicted in Fig. 1, is a customized Moodle instance, in order to preserve consistency with the learning management system in use in our University. To enable Moodle to adopt the social paradigm the SocialWall plug-in, (https://moodle.org/plugins/format_socialwall) was used. It turns the traditional Moodle course format into a social interface. Moreover, a Wiki activity as collaborative knowledge repository has been added. In particular, the Wiki enables users to co-create a complex web document, even without any knowledge about HTML, allowing them to gather pieces of knowledge and build a shared repository.

The SocialWall enables users to post messages, documents, links and any other kind of resources already available in Moodle. Users can express their like or comment the posts. All the social activities will appear on a timeline provided with some filtering tools.

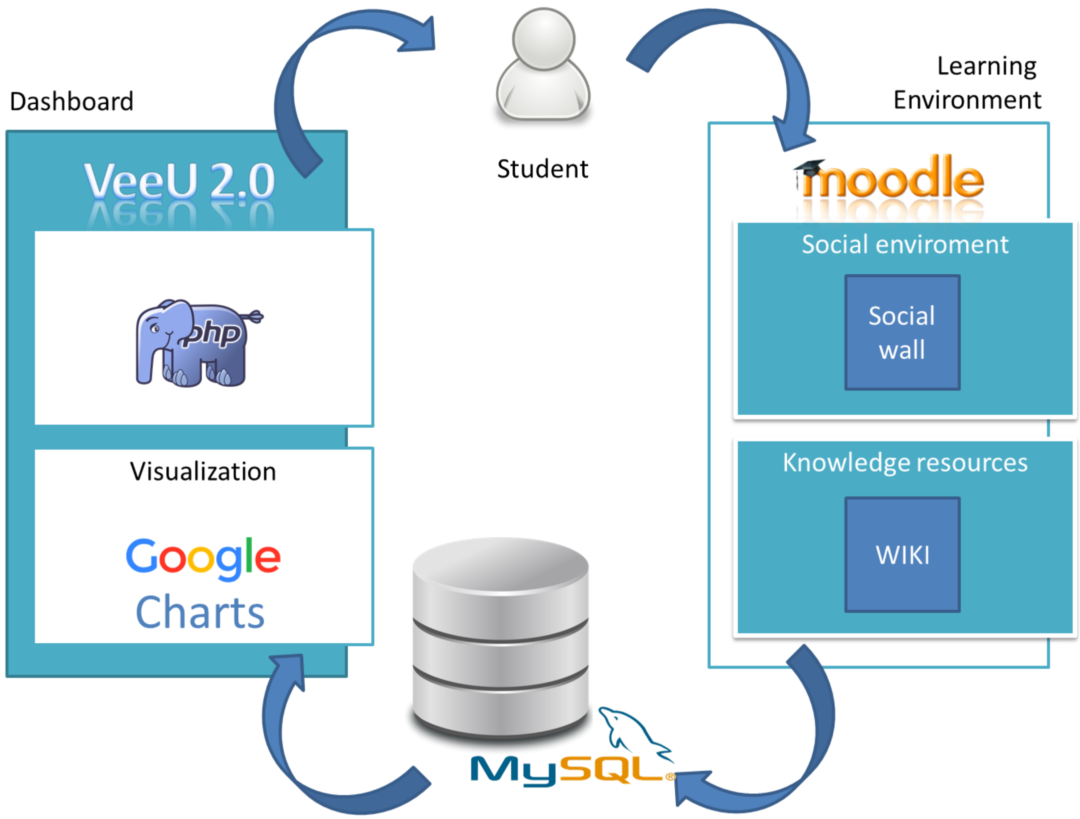


Figure 1. The Social Learning Environment

All the interactions occurring in the social learning environment are traced and stored in the Moodle database that is the data source of our dashboard. The trace-based approache, indeed, is one of the best method used in literature to detect users engagement, since they gather a lot of information without interfering with the learning activities [29] [30].

B. *Engagement indicators*

Engagement has been deeply studied in technology enhanced learning research from a wide range of perspectives: it is commonly recognized as a multidimensional and multifaceted construct but the definition of engagement is context dependent. In the specific context of social learning environments the engagement is strictly related to the activities performed in the community and to the users' participation in different kinds of activities.

First of all, it is necessary to classify the level of user's participation in the community, from a lower level, or peripheral participation [31], mainly consisting of reading resources and limited interactions with other users, to a higher level, or more active participation, consisting of activities such as writing comments on other users posts, sharing knowledge resources enabling a more responsible contribution to the knowledge of the community.

According to this perspective, in our research the engagement indicators were defined as follow:

- Passive interactions
    - Number of likes received by a post or a comment
    - Number of wiki pages read
- Active interactions
    - Number of posts published on the social wall
    - Number of comments shared on the social wall
    - Numbers of created wiki pages
    - Number of revised wiki pages

Furthermore, the next step is to identify the subject of interest of the user's participation. As defined in [32] the indicators of engagement are based on the participation both in social life of the community (interaction with other participants) and in the knowledge building activities (interaction with the knowledge resources).

According to the course structure we defined the engagement indicators as follow:

- Social Interactions
    - Number of posts published on the social wall
    - Number of comments shared on the social wall
    - Number of likes received on a post or a comment
- Knowledge Interactions
    - Number of wiki pages read
    - Number of created wiki pages
    - Number of revised wiki pages

All the indicators are calculated from the traces collected in the Moodle database: in particular standard Moodle log tables has been used for the indicators based on the Wiki while the SocialWall log tables have been used for the related indicators.

The basic indicators are collected at daily intervals and are then aggregated weekly or monthly according to the required level of analysis. The common aggregation level makes the indicators comparable over time and allows the timely evolution of engagement to be assessed. Then, as depicted in Fig. 1, the calculated indicators are presented through Google Chars using the most appropriate visualization as detailed in the next section.

C. Visualization

Our first goal was to visualize information about the entire class in order to understand the behavior and the level of participation of each student. Then we use our first pair of indicators to display the position of each student on a scatterplot. The students' position in the two-dimensional area is defined according to the number of Passive Interactions in a week on the x-axis and the number of Active Interactions in a week on the y-axis (Fig. 2). This visualization will provide a quick overview of the current status of the entire class, thanks to the scatterplot ability to display trends and relationship in a cloud of points [33]. Furthermore, outlier students will be easily identified but, in order to understand the reason of their poor participation, a detailed visualization is required.

To analyze the behavior of single student a visualization of engagement trends over time is needed. Thus, the second visualization (Fig. 3) depicts the details of a particular student in terms of trends and distribution. It is based on a linear visualization, a time series: two lines (one for each indicator) will display the trends of the engagement indicators over time for Passive and Active Interactions. In this case the indicators can be collected using daily, weekly or monthly intervals according to the required level of analysis.

Furthermore a pie chart will display the distribution of interaction in a selected time interval, between passive and active interactions, but also among the different components of each indicator (Number of likes received, Number of wiki pages read vs Number of posts published, Number of comments shared, Numbers of created wiki pages and Numbers of revised wiki pages).

After the analysis of the level of participation, the attention was focused on the kind of engagement. As a matter of fact, the more passive or active attitude of a student is not enough to evaluate the students' engagement.

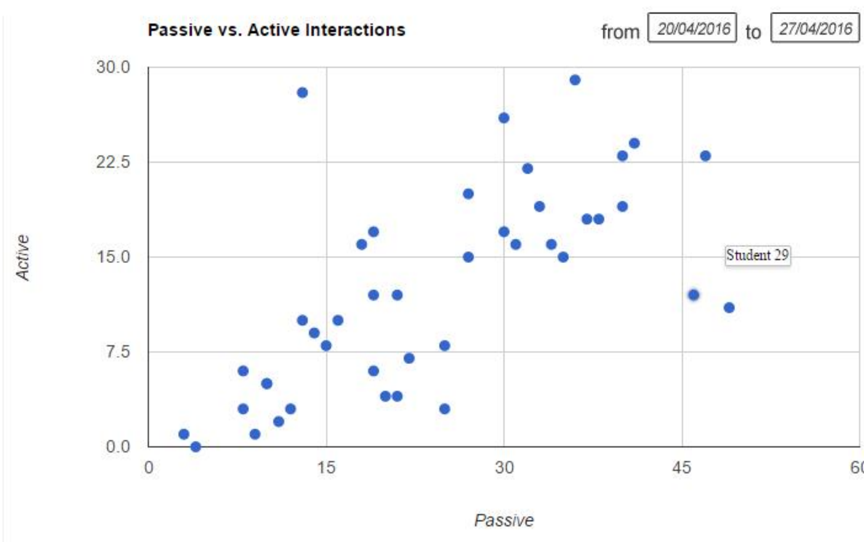


Figure 2. First visualization:
Scatterplot of Passive vs Active Interactions

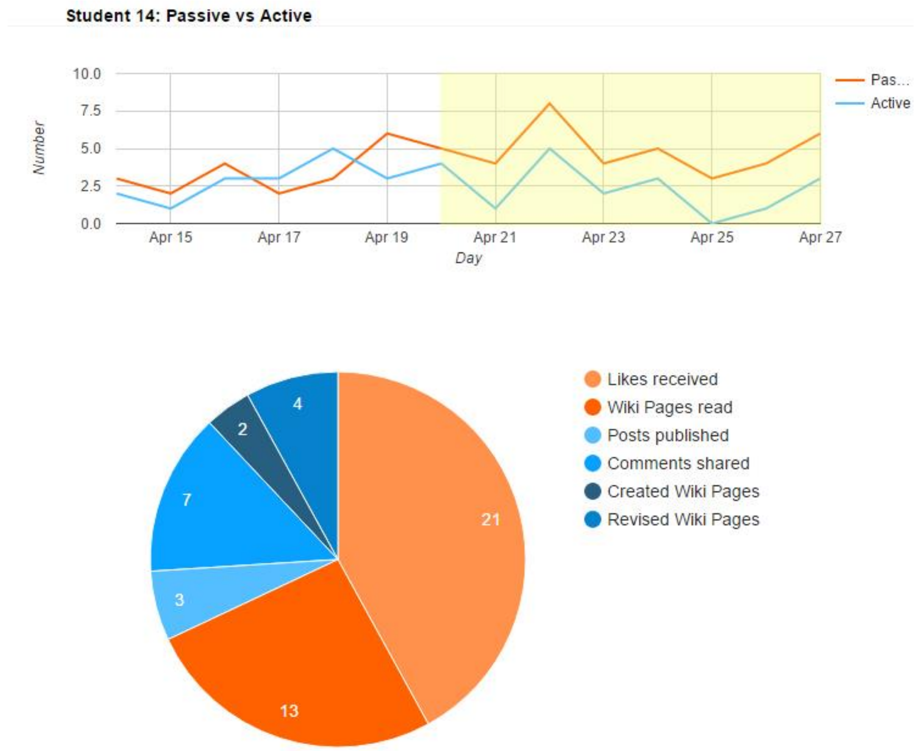


Figure 3. Second Visualization:
Trends and distribution of Passive vs Active Interactions for a single student

The third visualization (Fig. 4) is a scatterplot that, in this case, displays the students' position according to the number of Social Interactions in a week on the x-axis and Knowledge Interactions in a week on the y-axis. This kind of visualization will provide a quick overview about the subject of interests of student's engagement allowing the identification of purely socializer users or users actually involved in both Social and Knowledge Interactions.

A detailed visualization of a particular student will be useful in this case too. The fourth visualization (Fig. 5) presents the details of a particular student in term of trends and distribution with a time line and a pie chart built with this new pair of indicators.

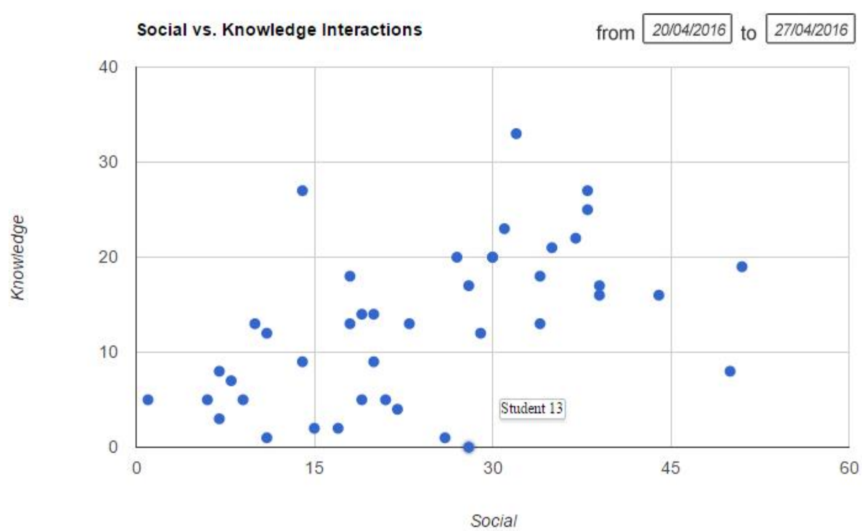


Figure 4. Third visualization:
Scatterplot of Social vs Knowledge Interactions

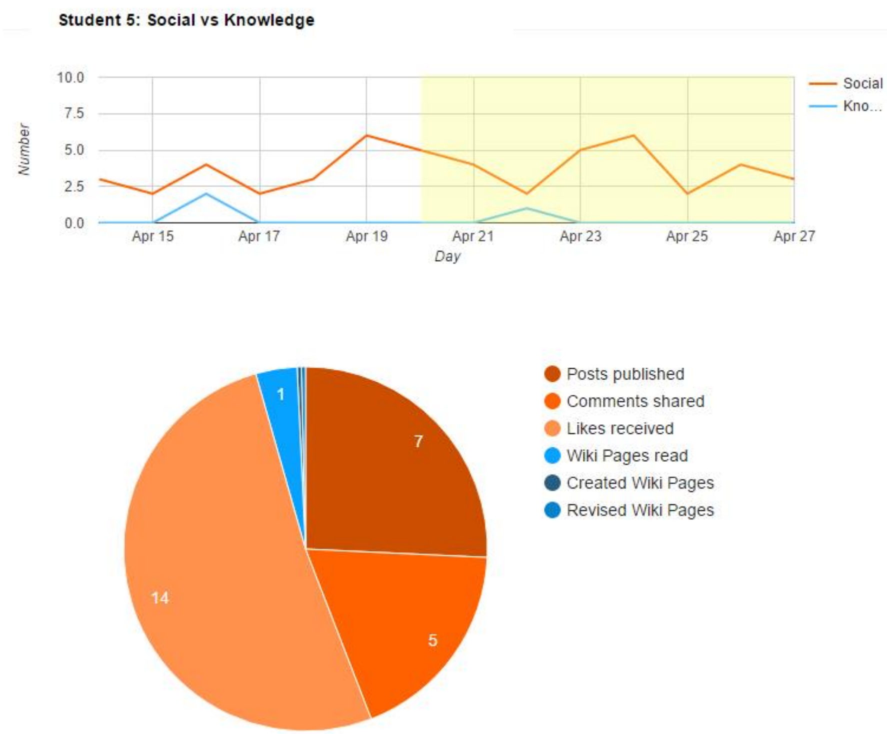


Figure 5. Fouth Visualization:
Trends and distribution of Social vs Knowledge Interactions for a single student

IV. CONCLUSION AND FUTURE WORKS

Our research aims to address the needs of information about the learning effectiveness in social learning environment, using engagement measures.

To this end, activity monitoring tools have become vital in distance education, thanks to the increasing availability of data collected by tracking the online activities. Information Visualization techniques, taking advantage of visual perception skills, are powerful tools to present a large amount of data by transforming traces in information. An accurate dashboard design is essential to provide students and teachers with a decision support tool that (1) promotes the awareness of what is happening in the learning process, (2) encourages personal reflection and achieving goals.

Data and reports usually provided by learning management systems are usually limited to measurement of activities and performances but there is a growing need to monitor less tangible aspects such as engagement in the social learning contexts. This is the main aim of the proposed dashboards, which is still under development. The approach adopted is an iterative process, which involves both researchers and social learning environment users. Researchers in the first design phase have involved teachers in order to define the most relevant indicators and to define the most appropriate visualization for the dashboards. The student were engaged in the pilot study, which involved two classes of undergraduate students from the Computer Lab courses held by the Department of Education, Psychology and Communication of the University of Bari Aldo Moro. Students are currently working in the social learning environment and the results of the beta test phase of the dashboard will be soon available. First results seem promising but further field test are required for both to measure the long-term effectiveness of dashboards in the learning process and to improve dashboard functionalities.

Future developments are under investigation. In particular, we are interested in monitoring the evolution of engagement indicators over time both for each single student and for the whole class. Static or dynamic time mapping approaches will be evaluated. Static visualization, such as small multiples, is more effective for analysis, whereas dynamic visualization, such as animation, was found to be more effective for presentation [34]. As stated by Chevalier [35], animations are commonly used (1) to replay history of events that occurred in a dynamic system allowing users to go back in time, and (2) to make activities and change visible at the same time, while keeping the user engaged thanks to the ability to attract user's attention, However, animation is not immune to fault such as higher cognitive load or perceptual effects of change blindness [36]. In order to overcome this drawbacks interactivity may be the key to overcome the difficulties of perception and comprehension [37] allowing users to explore time dimension, if needed, without distracting them from focusing on a single moment in time.

Furthermore, other indicators will be required to provide users with further insight of what is happening in the social learning environment. In particular, Social Network Analysis measures and social network visualization will allow us to make explicit the relationships between users; Discourse and Content Analysis will allow us to explicit the topic of interest [38]. Even in this case interactive visualization will be useful to analyze and present the available data: for instance a tag-cloud visualization of topics will allow users to refine engagements indicator based on a most relevant or on a specific topic.

REFERENCES

[1] J. Kennedy, "Characteristics of Massive Open Online Courses (MOOCs): A Research Review, 2009-2012.," *J. Interact. Online Learn.*, vol. 13, no. 1, 2014.

[2] T. R. Liyanagunawardena, A. A. Adams, S. A. Williams, and T. Rekha Liyanagunawardena, "MOOCs: a systematic study of the published literature 2008- 2012," *Int. Rev. Res. Open Distance Learn.*, vol. 14, no. 3, pp. 202–227, 2013.

[3] W. Admiraal, B. Huisman, and O. Pilli, "Assessment in massive open online courses," *Electron. J. e-Learning*, vol. 13, no. 4, pp. 207–216, 2015.

[4] P. J. Muñoz-Merino, J. a. Ruipérez-Valiente, C. Alario-Hoyos, M. Pérez-Sanagustín, and C. Delgado Kloos, "Precise Effectiveness Strategy for analyzing the effectiveness of students with educational resources and activities in MOOCs," *Comput. Human Behav.*, vol. 47, pp. 108–118, 2015.

[5] E. Pesare, T. Roselli, V. Rossano, and P. Di Bitonto, "Digitally enhanced assessment in virtual learning environments," *J. Vis. Lang. Comput.*, vol. 31, pp. 252–259, 2015.

[6] G. Siemens, "Learning Analytics : Envisioning a Research Discipline and a Domain of Practice," *2nd Int. Conf. Learn. Anal. Knowl.*, no. May, pp. 4–8, 2012.

[7] D. Bajzek, W. Brown, M. Lovett, and G. Rule, "Inventing the digital dashboard for learning," *World Conf. Educ. Multimedia, Hypermedia Telecommun. 2007*, pp. 1084–1092, 2007.

[8] G. Hwang, "Definition , framework and research issues of smart learning environments - a context-aware ubiquitous learning perspective," *Smart Learn. Environ.*, vol. 1, pp. 1–14, 2014.

[9] K. Verbert, S. Govaerts, E. Duval, J. L. Santos, F. Van Assche, G. Parra, and J. Klerkx, "Learning dashboards: An overview and future research opportunities," *Pers. Ubiquitous Comput.*, vol. 18, no. 6, pp. 1499–1514, 2014.

[10] S. Govaerts, K. Verbert, E. Duval, and A. Pardo, "The student activity meter for awareness and self-reflection," *Chi*, pp. 869–884, 2012.

[11] S. Govaerts, K. Verbert, J. Klerkx, and E. Duval, "Visualizing activities for self-reflection and awareness," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6483 LNCS, pp. 91–100, 2010.

[12] T. D. Loboda, J. Guerra, R. Hosseini, and P. Brusilovsky, "Mastery Grids : An Open Source Social Educational Progress Visualization," pp. 235–248, 2014.

[13] J. L. Santos, K. Verbert, S. Govaerts, and E. Duval, "Visualizing PLE usage," *CEUR Workshop Proc.*, vol. 773, pp. 34–38, 2011.

[14] J. L. Santos, S. Govaerts, K. Verbert, and E. Duval, "Goal-oriented visualizations of activity tracking : a case study with engineering students," *Learn. Anal. Knowl.*, pp. 143–152, 2012.

[15] J. L. Santos, K. Verbert, and E. Duval, "Empowering students to reflect on their activity with stepup!: Two case studies with engineering students.," *CEUR Workshop Proc.*, vol. 931, pp. 73–86, 2012.

[16] J. L. Santos, K. Verbert, S. Govaerts, and E. Duval, "Addressing learner issues with StepUp!," *Proc. Third Int. Conf. Learn. Anal. Knowl. - LAK '13*, p. 14, 2013.

[17] K. Verbert, E. Duval, J. Klerkx, S. Govaerts, and J. L. Santos, "Learning Analytics Dashboard Applications," *Am. Behav. Sci.*, no. February, p. 0002764213479363–, 2013.

[18] V. Rivera-Pelayo, V. Zacharias, L. Müller, and S. Braun, "Applying quantified self approaches to support reflective learning," *Proc. 2nd Int. Conf. Learn. Anal. Knowl. - LAK '12*, p. 111, 2012.

[19] G. Conole and K. Fill, "A Learning Design Toolkit to Create Pedagogically Effective Learning Activities," *J. Interact. Media Educ.*, vol. 8, pp. 1–16, 2005.

[20] M. May, S. George, and P. Prévôt, "TrAVis to Enhance Students' Self-monitoring in Online Learning Supported by Computer-Mediated Communication Tools," *Ijcisim*, vol. 3, no. May 2016, pp. 623–634, 2011.

[21] B. Yousuf and O. Conlan, "Enhancing Learner Engagement through Personalized Visual Narratives," *2014 IEEE 14th Int. Conf. Adv. Learn. Technol.*, pp. 89–93, 2014.

[22] B. Yousuf and O. Conlan, "VisEN: Motivating Learner Engagement Through Explorable Visual Narratives," in *Design for Teaching and Learning in a Networked World*, Springer, 2015, pp. 367–380.

[23] C. Coffrin, L. Corrin, P. de Barba, and G. Kennedy, "Visualizing patterns of student engagement and performance in MOOCs," *Proceedins Fourth Int. Conf. Learn. Anal. Knowl. - LAK '14*, pp. 83–92, 2014.

[24] R. F. Kizilcec, C. Piech, and E. Schneider, "Deconstructing Disengagement : Analyzing Learner Subpopulations in Massive Open Online Courses," *Lak '13*, p. 10, 2013.

[25] P. Di Bitonto, E. Pesare, V. Rossano, and T. Roselli, "Smart learning environments using social network , gamification and recommender system approaches in e- health contexts," in *Smart Education and Smart e-Learning*, Springer International Publishing, Ed. 2015, pp. 491–500.

[26] B. Schreurs, M. de Laat, C. Teplovs, and S. Voogd, "Social Learning Analytics applied in a MOOC-environment Authors," *eLearning Pap.*, vol. 36, no. January, pp. 45–48, 2014.

[27] R. Carrillo;, E. Lavouè, and Y. Prié, "Towards Qualitative Insights for Visualizing Student Engagement in Web-based Learning Environments Rubiela Carrillo The Dimensions of Student Engagement," *Proc. 25th Int. Conf. Companion World Wide Web*, pp. 893–898, 2016.

[28] S. Few, *Information Dashboard Design: Displaying data for at-a-glance monitoring*. Analytics Press, 2013.

[29] J. M. Kivikangas, G. Chanel, B. Cowley, I. Ekman, M. Salminen, S. Järvelä, and N. Ravaja, "A review of the use of psychophysiological methods in game research," *J. Gaming Virtual Worlds*, vol. 3, no. 3, pp. 181–199, 2011.

[30] P. Bouvier, K. Sehaba, and É. Lavoué, *A trace-based approach to identifying users' engagement and qualifying their engaged-behaviours in interactive systems: application to a social game*. 2014.

[31] J. Lave and E. Wenger, "Situated learning: Legitimate peripheral participation," *Learn. doing*, vol. 95, p. 138, 1991.

[32] E. Pesare;, T. Roselli, and V. Rossano, "Engagement in social learning: Detecting engagement in online communities of practice," in *International Conference on Applied Human Factors and Ergonomics*, 2016, p. in press.

[33] M. Friendly and D. Denis, "The eraly origins and development of the scatterplot," *J. Hist. Behav. Sci.*, vol. 41, no. 2, pp. 103–130, 2005.

[34] W. Aigner, S. Miksch, W. Muller, H. Schumann, and C. Tominski, "Visualizing time-oriented data-A systematic view," *Comput. Graph.*, vol. 31, no. 3, pp. 401–409, 2007.

[35] F. Chevalier, N. H. Riche, C. Plaisant, A. Chalbi, and C. Hurter, "Animations 25 Years Later : New Roles and Opportunities," pp. 280–287, 2016.

[36] Y. Albo, J. Lanir, P. Bak, and S. Rafaeli, "Static vs . Dynamic Time Mapping in Radial Composite Indicator Visualization," pp. 264–271, 2016.

[37] M. Tversky, B., Morrison, J. B., & Betrancourt, "Animation : can it facilitate ?," *Int. J. Hum. Comput. Stud.*, pp. 247–262, 2002.

[38] S. Buckingham Shum and R. Ferguson, "Social Learning Analytics," *Soc. Learn. Anal.*, vol. 15, no. 3, pp. 3–26, 2012.

Is e-learning ready for big data? And how big data would be useful to e-learning ?

Paolo Maresca[1], Andrea Molinari[2,3]

[1]Department of Ingegneria Elettrica e Tecnologie delle Informazioni (DIETI) – Università Federico II di Napoli (ITALY)

[2]School of Industrial Engineering and Management, Lappeenranta University of Technology (FINLAND)

[3] Department of Industrial Engineering - University of Trento (ITALY)

Abstract **The paper presents an overview of possible application fields of big data to the Technology-Enhanced Learning (TEL), with the many different facets this could imply. Many are the benefits for e-learning when approaching the collection of data, especially when e-learning is delivered in a life-long learning perspective. All these benefits could impact the future of eLearning, by revolutionizing the way we analyze and assess the eLearning experience. On one side, we present our experience in enriching the persistence layer of an LMS with a deeper log system on users' actions, in the perspective of collecting volumes of data compatible with big data tools and technologies, while highlighting some related issues. On the other hand we will deal with the first applications of cognitive systems that are responsible for catalysing the big data in analytics aimed at e-learning activities.**

Keywords: e-learning, big data, LMS architecture, cognitive systems

I. Introduction

Today, big data is one of the buzzwords that IT researchers and specialists use everywhere. So we have seen various "waves" of attention and "hype" on technologies and solutions related to Technology Enhanced Learning, but many of these hypes have revealed to be simply another buzzword and have not been consolidated. Today's buzzword that has been sweeping the world for a few years but has only just started appearing more commonly in eLearning is Big Data [10].

The term "Big data" refers to the huge amount of data coming from many different data sources, that become too large, complex and dynamic for any conventional data tools to capture, store, manage and analyse. Big data approaches and technologies interest many different application fields, and we will find "big data ...." in front of any data analytics tool. Most of the problems were related with data warehouses, technologies aimed at supporting decisions based on reconciliation of the different databases of the organization.

The research activities in big data are aimed to find faster and more scalable solutions to store and process all data collected, instead of using traditional data warehousing approaches that are expensive, hard to design and to implement. Big Data therefore introduced two issues: how to address the problem of storing such a large amount of data, and how analytics tools could be created for the problem of analysing these huge datasets.

The paper will present the experience of the authors in designing and implementing a mechanism that will generate flushes of data from the persistence of the virtual learning platform created for managing tens of thousands users in our region, both from Academy and Industry. After the design of the solution, we have implemented a traditional internal method for managing the first data source for big data, i.e., dedicated logs of the platform. With a potential of more than 70.000 users, the volume at the moment are still in the range of the high level of database application, with approximately ten of thousand users using every day the platform for their e-learning tasks.

The tasks are recorded both for educational and security reasons, being the logging tasks involved in the recording of mouse clicking and in SCORM(Shareable Content Object Reference Model)-based material logging. Being SCORM logging not enough for some educational paths, according to the known limitations of SCORM standard, we decided to implement a meta-SCORM service, where more SCORM packages could be used in an educational path. In this scenario, we have logs of the platform for clicks and users' actions, logs from the SCORM player, and logs from the educational path services.

In general, there are several elements of data gathering and manipulation inside e-learning platforms [11] that could push this application field towards tools and techniques typical of Big Data:

- Traditional Weblogs, being the application a web-based software;
- internal logs of usage of the platform, the so-called "digital breadcrumbs", that track the learner's journey throughout the entire learning experience;
- Mobile logs, where data about mobile learning actions are collected;
- Service logs, users' actions on the different elements of the platform like documents, forums, blogs, FAQ etc.;
- logs from the SCORM player, normally an external entity respect to the core services of

DOI reference number: 10.18293/DMS2016-042

the platform, with the records of the SCORM objects' execution;

- Tin-CAN API calls, in case the platform is connected or acting as a Learning Record Store (LRS).

Recently, a further set of new ideas in e-learning could increase the need of a structural change of Learning Management System (LMS) architecture towards approaches and technologies connected with big data:

- Massive Open Online courses (MOOCs), by definition a generator of high volumes of data
- Life-long learning, an old buzzword of e-learning that is still valid and interesting and, most of all, is another generator of big data, specifically along time;
- Serious games that will use materials inside the platform, thus generating a relevant dataset related with users' performances;
- Big Data will change the way we approach e-Learning design by enabling developers to personalize courses to fit their learners' individual needs [14].

The paper will present an overview of problems and opportunities related with the introduction of a big data approach to e-learning, both in the software architecture of the platforms and in the approach of e-learning stakeholders to this discipline. The paper is divided as follows: section 2 will present the potential application fields of big data technologies and approaches in e-learning. Section 3 will illustrate one specific example of big data in e-learning, i.e., the change in the persistence layer of a software platform for e-learning, while section 4 will present the use of big data technologies, specifically IBM "ask Watson" solution to e-learning settings.

II. BIG DATA AND E-LEARNING: THE POTENTIAL APPLICATION FIELDS

There are a number of reasons why big data may, very well, revolutionize the eLearning industry. First and foremost, it will allow eLearning Professionals to customize the eLearning experience to provide learners with more effective, engaging, and informative eLearning courses and modules. Big data also has the potential to impact the future of eLearning by:

a) Offering invaluable feedback.

While online surveys and discussions may offer feedback regarding the effectiveness of eLearning courses and modules, big data gives to eLearning professionals the chance to receive invaluable feedback that can be used to pinpoint where the learner, and the eLearning course itself, may need to be improved. For instance, if a learner is able to look at an analysis of where he/she fell short while taking the eLearning course, he/she can then figure out how to correct the issue moving forward. At the same time, if the online facilitator observes that the majority of the learners struggle with a particular module or assignment, he/she can make proper adjustments to improve learners' performance.

b) Allowing eLearning professionals to design more personalized eLearning courses.

If eLearning professionals are given the opportunity to know how their learners are acquiring information and what works best for them, in terms of content and delivery, then this will result in more personalized and engaging eLearning courses. As such, modules can be custom tailored to meet the individual needs of the learner, which will offer a high quality and meaningful learning experience.

c) Targeting effective eLearning strategies and eLearning goals.

Big data in eLearning gives us an inside look at which eLearning strategies are working and which ones aren't necessarily helpful in terms of eLearning goal achievement. For example, you can determine which eLearning courses are contributing to skill development and which eLearning modules or elements may be irrelevant. As such, you can then devote resources to the aspects that are working, so that the learners can receive the preparation they need to fulfill their career goals.

d) Tracking learner patterns.

With big data, eLearning professionals gain the rare ability to track a learner throughout the entire process, from start to finish. In other words, you can see how well they performed on a test, or how quickly they finished a challenging eLearning module. This helps you to pinpoint patterns that will not only enable you to learn more about the learning behaviors of the individual learner, but your learners' group as a whole.

e) Expanding our understanding of the eLearning process.

As eLearning professionals, it's essential that we learn as much as possible about how learners acquire and digest knowledge. Big data gives us the chance to gain an in depth understanding of the eLearning process and how the learners are responding to the eLearning courses we are delivering to them. We can even figure out which time of day they learn most effectively or which delivery methods allow them to retain information most efficiently.

This information can then be used to take our eLearning strategies to the next level. To make a long story short, embracing Big Data reshapes eLearning and it's a win-win situation for both the eLearning professionals and the learners.

III. One Testbed For Big Data Generation: The "Online Communities" Platform

The source of inspiration for big data analysis in e-learning is a virtual communities management platform entirely and autonomously created by our research team, starting since 1998. The approach used in most of the communities managed by the platform regards what is called "blended" approach, i.e. an e-learning mixed between frontal and online education training, asynchronous and synchronous, with online tutoring and frontal work sessions, all supported by our "Online Communities" system, of which, around the end of the 90s, the Faculty of Economics of the University of Trento has decided to adopt, followed by other public and private institutions.

Currently, "Online Communities" (OC from now on) is mainly used outside the university campus, serving approximately 50.000 users from different public and private customers against approximately 15.000 students in our University [16].

OC is a dynamic web application, based on the metaphor of the virtual learning community, which ensures the cooperative organization of work in groups of users called "Community". A virtual community is defined as a space of communication shared by a group of people, not only related to educational aspects. Every community has at least one coordinator and participants that are not anonymous. It is natural to imagine a virtual community as an aggregation of individuals made possible thanks to computers; an extension in the virtual learning environments is the class in which the courses take place.

The system is designed from the ground up within the Laboratory of Maieutic working group - Department of Industrial Engineering of the University of Trento, and is able to support the needs of a broad group of users (teacher, student, tutor, lecturer, external consultants, supervisor, dean, counsellor, secretary, board member etc.), customizable within the context in which the system is used (for example in a business organization we will have different roles respect to the university, as president, secretary, director, administrative, board of director etc.).

The participants in the system are not anonymous, and have a number of roles; each role brings with it specific rights and duties. Therefore, the enrolled users participate in a series of communities, fulfilling different roles in each of them. The communities are also characterized by a series of events that correspond to the active involvement of members in different moments. OC was released outside of test environments in 1998 as a working prototype, then reached through different evolution steps its maturity in early 2005, but counting on a long experimentation on a limited number of courses started at the end of 1998.

Since September 2005, the system is in operation at the University of Trento, involving tens of thousands of users, and has become an everyday tool for the teaching of many teachers. It represents the technology infrastructure used officially by some faculties of the university of Trento for testing and supporting new forms of teaching based on the techniques and the methods of e-learning. In 2007, the Autonomous Province of Trento decided to adopt it as its platform to deliver e-learning for its approx. 20.000 users. Then the Chamber of Commerce of Trento and the Chamber of Commerce of Bolzano adopted the platform and extended it to the affiliated enterprises, where we have just for the Province of Trento approx. 55.000 enterprises with the respective users.

The platform enables the members of any community to collaborate using multiple computer services (blog, wiki, chat, video conferencing, forums, file sharing, diaries , diaries, etc.). These services are used in areas collaboratively, both in the training, given the apparent closeness between the two environments. It 'also clear that, talking about continuing education and lifelong learning, the boundary between training and collaboration is increasingly being blurred when dealing with adults involved in the workplace. The logic of a virtual community platform considers any combination of people, brought together in a virtual place (hence "virtual community") for various purposes.

From our analysis, at the moment the interaction of users generate 2GB of data per month only for the users' actions log, just for 2.000 users/day. It's clear that the overall picture could become much more compelling for any software platform where the collection of data could stimulate and support DSS from the top management of the institution, or where hundreds of thousands participants follow intensive MOOCs.

Most of this data are coming from the first data source we used in our experimentation due to its affinity with big data data sources, the aforementioned "Actions". It collects all data coming from users' interactions with any OC object or service. It is a sort of "sensor" introduced inside the source code of the platform in any place is needed the software to capture an "action" from the user interface. This is of course a relevant enrichment of the logs recorded by the web application server, and has been used for many different purposes.

Due to volume issues, the system at the moment is blocked on collecting only some types of events, to a certain granularity defined by the system administrator. This choice has not been a design choice, but a performance-related one. From the early experimentations, it was clear that the amount of data could have compromised the capacity of the DBMS to stand data acquisition pace and volumes: a typical "Velocity" and "Volume" big data problem.

Tasks are recorded both for educational and security reasons, being the logging tasks involved in the recording of mouse clicks and SCORM-based material actions. Being SCORM logging not enough for some educational paths, according to the known limitations of SCORM standard, we decided to implement a meta-SCORM service, where more SCORM packages could be used in an educational path, mixing them with other services provided by the platform (like Wikis, FAQs, forums etc.) in a unique view called "educational path". In this scenario, we have logs of the platform for clicks and users' actions, logs from the SCORM player, and logs from the educational path service.

From the potential amount of data generated by these actions, we started to imagine our big data approach to the platform.

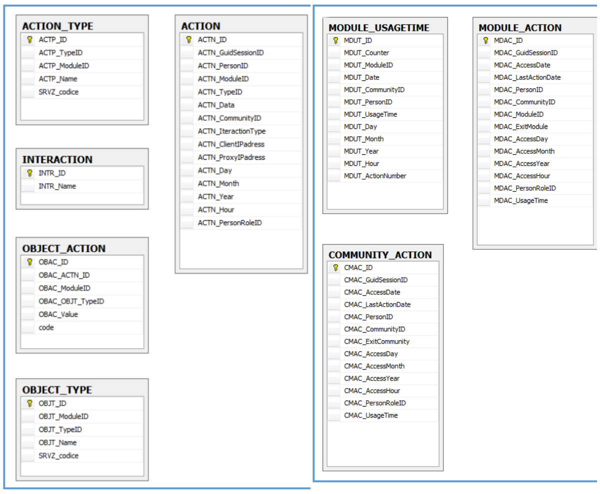


Figure 1. data structures for big data analysis in "Online Communities"

IV. Big Data As Support To Learning Managemement And Learning Process

One of the domains in which you can take on the challenge of big data and the cloud is the *management of learning processe*s as well as the *customization of the learning managemen*t. The *customization of learning processe*s can be accomplished employing the cognitive system as the IBM Watson [2] according to a paradigm known as to ***ask Watson,*** with which you establish a triage *Watson-student-teacher* that will be explained in detail in the following paragraphs.

Thanks to the use of cognitive computing, classes become more intelligent, not only for the content they make available to the students but for the possibility of using student's data to figure out how to grow the class. The observation period may be as long as much as they are in the school and beyond[3].

These systems can also help teachers deal with the problems of the students such as school dropout, credits recovery etc., thus allowing a process that aim at learning *customization*. The paradigm is innovative and is based on the fact that the teacher asks the cognitive system to "talk" with the student in order to understand which are the strengths and weaknesses to be reported to the teacher who then will decide which course materials should be delivered to overcome the identified learning problem. This feature is known as ***Ask to Watson*** and represents the aforementioned ***triage*** *student-Watson-teacher* which represents the true individual learning customization.

As an example, you can ***ask Watson*** to discover the learning mode of the student (kinaesthetic, visual, auditory) in order to deliver the appropriate training material. Or you can ***ask Watson*** "to prepare a small subclass for the Olympics of mathematics or computer science" and then provide the material and proper training to achieve particular objectives challengers. But the most important thing is the interaction with Watson that occurs in spoken language.

To date, Watson interacts in English, Japanese, Spanish and Portuguese, it learns from the students using the features outlined in the previous paragraph. In addition, the teaching material is densely developed with HTML tags and hypertext, so that you do not need to do complex searches and the concepts are often proposed using games and flowing into the domain appropriate to young adults.

There are many existing projects in this area around the world. Georgia's Gwinnet County public schools, is the 14th most populous district of the USA and is one of the first college that has pioneered the use of Watson. Watson has the task of "*identifying similarities in how students learn and predicting performance and learning needs",* while "*specific content and teaching techniques can be aligned to each of the district's 170,000 students to Ensure the best learning experience"*. In particular, the paradigm is quite complex and is constituted by recursive cycles that consist of the following 6 steps:

i) Watson identifies weaknesses and strengths of every students;
ii) Watson recommends behaviour and contents for students aligned to their skills and learning styles;
iii) Teacher selects appropriate learning path and Creates a progression plan for student;
iv) Students consume the recommended content from the material plan;
v) Teachers monitor the progress of the students and adjust the plan;
vi) Teachers use Watson to identify student skills attainment aligned to the standards defined.

More specifically, it is interesting to find out how they interact between Watson, the student and the teacher through the services it offers. The interaction occurs in spoken language and consists of three cognitive services: I) **Ask**, II) **discover**, III) **decide** delivered in sequence. In the first the student asks Watson to guide him, in the second Watson teaches the student how to derive the answers from the questions asked, in the third the teacher sees the results and delegates Watson for the adaptation of the training course. This mode of *assistance* is very promising and is bearing much fruit.

Other challenges have been launched in this field especially in the analysis of data and analytics [1][6][7]. For example, Wichita State University uses advanced analytics to predict potential students chances of success: 15% boost in registration. Hamilton County, Tennessee's Department of Education uses predictive analytics to improve student achievement causing the Following results: 8% increase in the graduation rate to 80% and 25% reduction in the annual drop rate. Seton Hall has used integrated marketing optimization solutions analysing social media to understand how students move through the recruitment process: 13% enrolled increase.

The novelty of this approach lies in the use of natural language (English, Spanish, Portuguese and Japanese) that allows Watson to discover the student's difficulties and to report them to the teacher who chooses which educational materials to provide, and so on until you reach the training course objectives that have been established for that student. Training materials can be chosen on the basis of intellectual channels of young people and can be provided on the basis of the receptive attitudes of the students (auditory, visual or kinaesthetic).

At this stage, the interaction through natural language helps a lot especially if you are using two essential functions of cognitive system that are called speech to text and text to speech (see fig.2). The first transforms written text into speech, the second performs the inverse function. Based on these features, for example Sundararajan and Nitta [12], designed and realized a tutoring system for K12 students, intensively using interactivity, automatic generation of questions, and learning analytics. It is worth noting that these features, as all the others, are available through Platform-as-a-Service (PaaS) bluemix IBM and the access for Universities is a free of use under the IBM academic initiative agreement for six months.

Watson Ecosystem is one of the largest research organizations in the world, only in the IBM business intelligence industry spends 6 Billion of dollars for years since 5 years. Watson is accessible through a cloud open platform called Bluemix (see fig.2), has spread to 35 countries, with 18,000 programmers who work there and 500 people making their business all over the world. Watson currently has more than 30 services, 15 underlying platform technologies, and thanks to bluemix, enables the output of its API to anyone who requests it.

Figure 2. Watson and their services (boiler plate) inside Bluemix Paas

The big data analytics is the key of managing the learning processes [13][14]. This allows us to understand and observe the students careers during their training to provide them with assistance and suggestions about the composition of university curricula, management training curricular and extracurricular, insights and the advice to acquire specific skills. New generations of (special) applications that precisely interact with a cognitive system (and therefore are named ***cognos***) are starting to appear. An interesting experience of using cognitive computing in educational settings is the Watson Student Showcase [8] organized by IEEE jointly with IBM. Students were challenged in a competition where they had to develop cognitive apps, using the cognitive computing services included in Bluemix.

The cognitive computing and the internet of things (IOT) will be the real innovation of the future. This will have the disruptive implications in the way of teaching [9]. If you only think about how our laboratories will be managed and maintained, with everything on cloud and with students interacting with equipment and robots that will in turn interact with other things using simple interfaces and often through the voice. all this can be achieved by developing a new generation of cognitive app that uses the voice and therefore might call cognitive app (Cognos).

Cognos learn and communicate with other machines making it easy to manage the interaction between man and machine and from machine to machine. Consider the impact of all this on our teaching laboratories: electrical and electronic measurements, computer science, robotics etc. Labs may be automated, students will be able to use them at anytime from anywhere, without limitation of time. An approach of this type will revolutionize education, teaching and the contents. Of course cognitive computing as Watson and other can make this already possible [7].

Another interesting side is the bureaucratic management of teachers and students. For the management of students, the observation is not only limited to the period of University residence, but can be extended before (in high school) and after (in employment) in order to have a traceability of the careers of the skills and knowledge acquired. All this is achieved through the management of big data and through the management of platforms dedicated to this type of objective. Again cognitive systems [5] well compete through the analytics. Even the management of teachers is interesting through analytics, and more generally, many problems can be managed through analytics foreshadowing a smarter Universities [4].

The analytics can also be extended to networks of people who share a common goal. The ***nose of knowledge*** helped a lot in building a network of expertise and knowledge [7]. But this knowledge is often limited to students in a class network. This causes the potential of the group remain limited. If we were interested to boost the potential of this network group should be extended to accommodate the students' knowledge and skills from other classes, other universities geographically located distant from us. Precisely in this regard you may develop a ***nose of knowledge cognos app*** that extends the capabilities of another app developed to the Ohio state university.

In fact The **Ohio State University** created a web application for the Watson University Competition called the "**YouDoU**" [15] project that helps Ohio State students find activities based on their personal preferences using Watson technology. An extension of personal preferences are also represented by the skills and knowledge acquired during their university studies that everyone wants to pool in a workgroup. This extension can be managed by ***nose of knowledge cognos app.*** Many analytics can be used. Analytics can be used to measure the reputation that each student acquires in each group, by virtue of the activity that takes place during the collaboration. This way we can dynamically reconfigure the network by combining the elements that achieve the highest score, excluding those that bear a lesser reputation of a certain threshold, etc.

Finally analytics can also be applied on Italian University's admission tests to identify the University course credits that each student must mature and then follow him in his career and give him a concrete and effective customized training support.

V. CONCLUSIONS

In this paper we presented some application scenarios of big data on e-learning. More generally, the authors think that the ability to think, learn and understand the world around them will be essential to compete in the near future both for universities and companies . All this will be possible by the use of big data. We are surrounded by big data, and it is natural to use them to do business through analytics. In Universities, it is natural to use big data to manage learning processes in an intelligent way and to help individual students to improve their skills.

From the perspective of companies, we are facing a very strong disruptive technology that will transform everything, i.e., cognitive computing. Technologies related with analytics, cloud , mobility, security are very important, but one question arise to which we have to find an answer: when will all companies be digitized enough to compete in the market and win the competition?

The point is that the process of digitization in a company is not among its primary goal, but constitutes the premises for further digital revolution: the cognitive IOT. The two drivers of this new technology will be invisible and cognitive data. Eighty percent of data is invisible and is not understandable, "obscure" so to say, because it has not a structure but is also stored somewhere. This includes video, music, news articles, research reports, social media posts, enterprise data system etc. The second driver is even stronger and is represented by cognitive that no longer requires programming.

Cognitive systems in fact require the use of spoken language and there are currently many existing applications, although at an experimental level, on sails, supply chain, research, child (with *dyno* cognos), bluematch (putting together positions, persons and curricula of each one in order to reach specific project objectives), education , cooking recipes, medicine, meteorology, etc.). The difference between the cognitive and the current systems is that systems understand, reason and learn. It 'a revolution in technology, business and teaching.

The revolution in teaching will be even stronger because it will change the paradigm, the educational materials and the laboratories, and this in turn will imply the change in the role of the teacher, that will have a co-coach role assisted by cognitive systems. The support to this new paradigm of teaching and learning will cause a circular mechanism, where more and more data will be needed. So the sooner the e-learning world will realize and implement big data inside their technologies and processes, the better for the whole e-learning movement.

REFERENCES

[1] E. AbuKhousa and Y. Atif (2014), "Big learning data analytics support for engineering career readiness," Interactive Collaborative Learning (ICL), 2014 International Conference on, Dubai, 2014, pp. 663-668.

[2] Banavar, G.S. (2015), *Watson and the era of cognitive computing,* in Proceedings of the 2015 IEEE International Conference on Pervasive Computing and Communications (PerCom), 95-95.

[3] Byrd, G. (2016), IEEE/IBM Watson student showcase, IEEE Computer, 49 (1), 102-104.

[4] Coccoli M., Guercio A., Maresca P. and Stanganelli L. (2014) "Smarter Universities: a Vision for the Fast Changing Digital Era", in Proceedings of 20th International Conference on Distributed Multimedia Systems (DMS2014)

[5] Coccoli M., Maresca P., Stanganelli L. (2012) "Cognitive computing in Education ", Journal of e-Learning and Knowledge Society (Je-LKS),Vol. 12, n. 2, May 2016, pp. 55 – 69.

[6] Ferguson, R. and Buckingham Shum, S. (2012). Social Learning Analytics: Five Approaches. LAK 2012. 23--33. DOI: 10.1145/2330601.2330616

[7] Gašević, D., Dawson, S., & Siemens, G. (2015). Let's Not Forget: Learning Analytics are about Learning. Techtrends, 59(1), 64--71. DOI: 10.1007/s11528-014-0822-x

[8] Holtel, S. (2014), *More the end of information over ow: how IBM Watson turn upside down our view on information appliances*, in Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-I.T), 187-188.

[9] Coccoli M., Maresca P., Stanganelli L., and Guercio A. (2015) "Teaching Computer Programming in a Platform as a Service Environment", in Proceedings of 21th International Conference on Distributed Multimedia Systems (DMS2015)

[10] Riffai M. M. M. A., Duncan P., Edgar D. and Al-Bulushi A. H., "The potential for big data to enhance the higher education sector in Oman," 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC), Muscat, Oman, 2016, pp.1-6. doi: 10.1109/ICBDSC.2016.7460346

[11] Siemens, G. & Long P. (2011) Penetrating the Fog: Analytics in Learning and Education. EDUCAUSE Review 46 (5), 2011, pp. 30-38.

[12] Sundararajan, S.C. and Nitta, S. (2015), Designing engaging intelligent tutoring systems in an age of cognitive computing, IBM Journal of Research and Development, 59(6), pp. 1-9.

[13] Vozniuk, A., Govaerts, S., & Gillet, D. 2013. Towards Portable Learning Analytics Dashboards. ICALT 2013, IEEE, 412--416. DOI: 10.1109/icalt.2013.126

[14] West, D. 2012. Big Data for Education: Data Mining, Data Analytics, and Web Dashboards. Governance Studies at Brookings, pp. 1-10.

[15] Youdou, https://www.youtube.com/watch?v=LiOI07dQVTI

[16] Colazzo, L.; Molinari, A.; Villa, N.; , "Collaboration vs. Participation: The Role of Virtual Communities in a Web 2.0 World," Education Technology and Computer, 2009. ICETC '09. International Conference on , vol., no., pp.321-325, 17-20 April 2009

Supporting Mobile Development Project-Based learning by Software Project and Product Measures

Rita Francese[1], Carmine Gravino[1], Michele Risi[1], Giuseppe Scanniello[2] and Genoveffa Tortora[1]

***Abstract*— Project-based learning is a kind of learning activity which has great educative effect, but which presents also several issues. In particular, if we consider an university course that requires the design and the implementation of a software project, may be difficult to estimate the number of hours that a team of students has to take to accomplish that project. There is the risk to underestimate the project (too difficult) or to overestimate it (too easy) with respect to the other projects of the same course and the amount of foreseen work hours. In this paper, we present the experience we gained in the adoption of Software Project and Product Measures for addressing the project size of projects performed during a Mobile Application Development course for Computer Science students at the University of Salerno. The course foresaw a project work conducted by students organized in teams. The goal of the project work was to design and develop an Android-based application with back-end for smart devices. Software estimation project measures are applied to some metrics extracted in the requirement analysis phase to get an estimation of the effort in terms of man/hours and consequently to adjust the project size by adding/reducing requirements. The metrics extracted from the projects of academic year 2013/14 have been used in the successive year for estimating the project effort and intervene on the project size variables.**

I. Introduction

Software development projects may often be late and overrun time and cost [1]. These can have a dramatic impact in case of Project-Based-Learning (PBL) activities, where it may occur that a student project is late or requires much more work with respect to the estimation. Effort understimation may produce the failure of the project and the team members risk to not pass the exam, while effort overstimation may produce a project not corresponding to the teacher expectations. The final evaluation can be excessive if the teacher does not correctly evaluate the project size, or students can have poor results, while they could have produced more with appropriate requirements.

Software estimation problems have been largely investigated in the literature. In particular, in [2] the problem of predicting software project and product measures in the domain of Android development has been addressed considering as sample mobile applications produced by students during a mobile application development course.

In this paper, we try to reduce the gap between the amount of hours required to develop a project and the real project effort in the domain of PBL for Android app development. We decided to try to estimate in the earlier phases of the projects the required effort and to resize the project requirements according to this suggestion. In particular, we exploited the results we got in [2], where we demonstrated that requirement analysis measures can effectively employed to estimate software project and product measures of a mobile app and estimations can be done early in the software development process. The metrics extracted from the projects of academic year 2013/14 have been used in the successive year to intervene on the requirements to adjust the project size.

The paper is structured as follows: in Section II we discuss the background concerning PBL and Software Estimation; Section III describes the learning experience related to the considered Mobile Application Development course; Section IV summarizes the effort estimation model we adopted and Section V describes how it is used to estimate student efforts and the results from its application. Finally, Section VI concludes the paper.

II. Background

Project-Based Learning (PBL) is a learning approach that structures learning around projects [3]. It involves both constructivisme [4], a learning theory in which learners actively construct their knowledge, and team-based learning [5], in which the learning activity is conducted by a team of people which collaborate to pursuit a common objective.

In particular, Vigotsky [6] puts in evidence the relevance of the social context and the interaction among pairs to learn when problems have to be solved. This overcomes the idea of learning activity where students passively absorb knowledge from their instructor. When student are motivated to cooperate they can be successful team member in real industry environments [7]. However, team work is not only work together, it requires the team members to plan their projects activities, to monitor the project progress, and to disciplinarly manage their work [8].

PBL let students cooperate to solve together problems typical of the job world. The involvement of students is in general higher than traditional classroom activities.

The adoption of PBL in Computer Science courses is growing [9], [10], [11], [12], [13] because it enables the students to acquire soft skills such as to be able to work in group as a team, to respect deadline, to take ones own responsibility, to be able to communicate. All these skills are considered very relevant for organizations. PBL promotes principles, methods and procedures similar to the ones adopted for developing real software products [14]. During the various project phases documentation is produced, following specific standards. This implies that students do not

[1]University of Salerno, Italy {francese, gravino, mrisi, tortora}@unisa.it
[2]University of Basilicata {giuseppe.scanniello}@unibas.it

DOI reference number: 10.18293/DMS2016-045

have to perform only coding activity, but also to acquire analysis and writing skills [10].

In [15] we described the learning experience related to the Mobile Application Development course, summarized in Section III. During this experience we collected several product and process metrics useful to assess the project and process quality. From this experience we gained the conviction that there was the need of following a project management approach which takes into account effort estimation for assigning correctly dimensioned projects to students' teams.

In the literature, estimation approaches have been classified in different way, see for example [16]. Generally, they belong to one of this types: *(i)* expert estimation, as software experts provides an estimate on the base of his experience; *(ii)* formal estimation model, a mathematical model is created on the base of historical data to quantify the effort; and *(iii)* a combination of the other two approaches. The estimation produces software measures which can be exploited for project management purpose.

Research studies put in evidence that the best approach does not exist, but the goodness of the results often depend on the context and the application domain [17], [18]. Also the development technology may influence the estimation accuracy [19].

Software estimation effort has been adopted in [20] for adjusting the size of database-oriented student projects, which followed a traditional waterfall development model. The prediction approach exploited a modified function point counting tables and use cases.

In this paper we apply the effort estimation model for mobile application development presented in [2], which is based on the metrics extracted from the projects of academic year 2013/14. This model has been conceived using the dataset of the experience presented in [15] and is detailed and discussed in Section IV.

III. The Learning experience

The Mobile Application Development (MAD) course is given at the University of Salerno since the academic year 2011/12. The course focus on the design and development of Android mobile applications. The course lasts 48 hours, divided in 24h lectures and 24h laboratory. It is estimated that for each hour of lab students have to perform at home two hours of study to implement their project.

In this paper we refer to two successive edition of the course: the first occurring in the spring of 2014 (described in [15]), and the second in the spring of 2015. The data got from the first edition have been used to develop an effort prediction model for the development of mobile applications, while in the second one we used the model to address the work of the student trying to get a balanced project work.

When the course begins, the project work is accurately motivated by each team by providing a *Project Proposal* document. The teacher evaluates this document, in which the objective of the project, the analysis of existing similar application, and a preliminary description of functional and non-functional requirements were provided. In particular, the teacher imposed the following non-functional requirements: the operating system is Android, the app had to interact with a remote server, communicating data through JSON; native device functionalities have to be exploited, including maps, GPS, sensors, phone call and SMS. Also the device rotation has to be managed and the app has to use SQLite to store data on the device. Games were admitted if they exposed back-end functionalities, such as account management, multiuser support, bonus management, and the upgrade of the app.

The team composition was freely chosen by students. We did not decide to assign randomly team members because the students had previous experience of project work in several courses and, at the last term, they know which were the classmates more appropriate for them. The students were asked to use GitHub[1] for the management activities. The lecturer created a GitHub account for each team of students. The templates of documents to be released at the end of the various development activities were made available in the GitHub repository of each team.

We established a schedule for each team of students. They were informed about deadlines and deliverables. The first deliverable was the project proposal, while the second the Requirement Analysis Document (RAD). Successively, the participants had to deliver the different releases of their mobile app and the final version of this app. To write the RAD, participants were asked to follow the template by Bruegge and Dutoit [21]. A RAD is used to document requirements elicitation and analysis. As for requirements elicitation, the software engineer specifies functional and non-functional requirements. Functional requirements are expressed as free-form text (a summary of functionality to be implemented) and then in terms of use cases narratives and use case diagrams. As for requirements analysis, the software engineer specifies object and dynamic models of the software. Each identified object of the problem domain is textually described, while relationships among objects are illustrated with class diagrams. Attributes and operations of problem domain objects are detailed only if needed. The behavior of these objects is documented in terms of sequence diagrams and state machine diagrams. These kinds of diagrams are exploited to specify complex behaviors of either use case or single objects, respectively.

We asked the students to follow an incremental prototyping development approach. The students were also asked to show app prototypes to the MAD lecturer before the conclusion of their project. At the end of the course a competition named *App Challenge* was conducted to whom participated members of International IT industries. This produced a good competition spirit among participants. The IT managers judged the final products of good quality and very near to real apps available on the market.

More detail on the teaching experience of the first edition are described in [15].

[1] https://github.com

TABLE I

VARIABLES DENOTING INFORMATION FROM RADS

Measure	Description
FR	Number of functional requirements
Act	Number of use case actors
UC	Number of use cases
Cla	Number of classes
SD	Number of sequence diagrams

TABLE II

VARIABLES DENOTING INFORMATION FROM SOURCE CODE OBTAINED BY THE UNDERSTAND TOOL

Measure	Description
McB	McCabe Cyclomatic complexity
Classes	Number of classes
Files	Number of files
Methods	Number of methods, including inherited ones
NL	Number of all lines
LOC	Number of lines containing source code
CLOC	Number of lines containing comment
STM	Number of statements
DIT	Depth of inheritance

IV. THE ESTIMATION MODEL

The estimation model has been created by analyzing the projects of the academic year 2013/14. In particular, we have considered the projects performed by students during the MAD course for Computer Science at the University of Salerno. Participants were originally 57 and were arranged in 27 teams. Data for 4 teams have been discarded due to incompleteness issues (e.g., lack of class diagrams or functional requirements) in the RADs the members of these teams produced. In the following of this section we describe how the model was built.

A. Variables

The dependent variable of interest for this discussion is the *Effort*, and can be computed as *the total effort to develop a mobile app expressed in terms of person/hour*.

In [2] initially were proposed two sets of independent variables. The first set, reported and described in Table I, collects variable obtained from the RAD or from the requirements, while the second set in Table II, collects variables from the source code gathered by exploiting the Understand[2] tool. These two sets of variables are useful to compare prediction accuracy of software measures obtained from RADs (RAD or requirements measures, from here on) against the accuracy of predictions obtained with measures gathered from source code (simply SC measures, from here on).

To explain how metrics have been collected, we provide an example of a project among those considered in our data analysis, *Archeotour*. Students of Archeotour team developed an Android application that provides information on archaeological sites and suggests tours considering the user's position and interests. Weather information is shown on demand.

The number of functional requirements (FR) of Archeotour is 8. It is obtained by counting requirements listed in the functional requirement section of the RAD. As an example, *"Select the site on the map and show its description, history and pictures"* is a functional requirement. A functional requirement can correspond to more than one use cases (UC). For example, the mentioned functional requirement is associated to the use cases *Show Site* and *Show Map* in Figure 1 showing one of the use case diagrams of the project. In this diagram, the number of use cases is 6, while the number of actors is 2. The number of actors (Act) is computed by counting the different actors that appear in all use case diagrams in the RAD. Similarly, the number of classes (Cla) is computed by counting classes in the class diagram of the same RAD. For example, the class diagram of Archeotour shown in Figure 2 is composed of 24 classes. The number of sequence diagrams (SD) is obtained by counting how many of this kind of diagrams have been specified in the RAD.

Some descriptive statistics (i.e., minimum, maximum, mean, median, and standard deviation values) of the independent variables are shown in Table III. For the dependent variables, descriptive statistics are also reported. We have also graphically shown the values for our dependent variables by the boxplots in Figure 3.

B. Estimation technique

The estimation technique adopted in [2] is StepWise Linear Regression (SWLR) technique [22], which explores the relationship between a dependent variable and one or more independent variables, providing a prediction model described by a linear equation:

$$y = b_1x_1 + b_2x_2 + ... + b_nx_n + c$$

where y is the dependent variable, $x_1, x_2, ..., x_n$ are the independent variables, b_i is the coefficient that represents

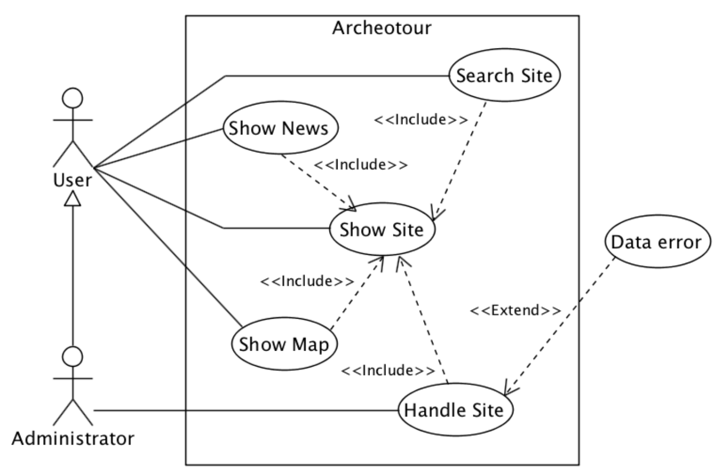


Fig. 1. A use case diagram of the Archeotur project.

[2] https://scitools.com

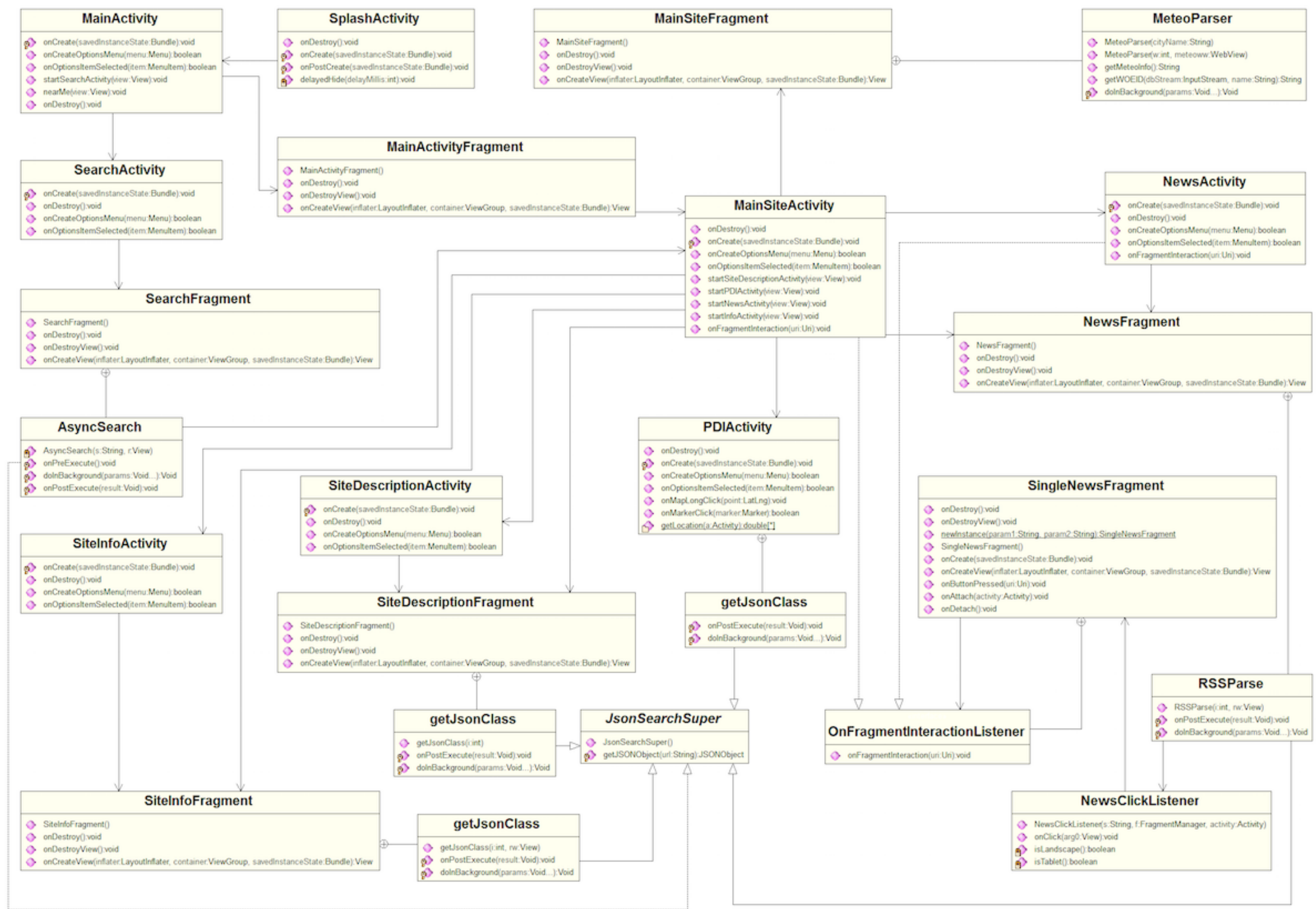


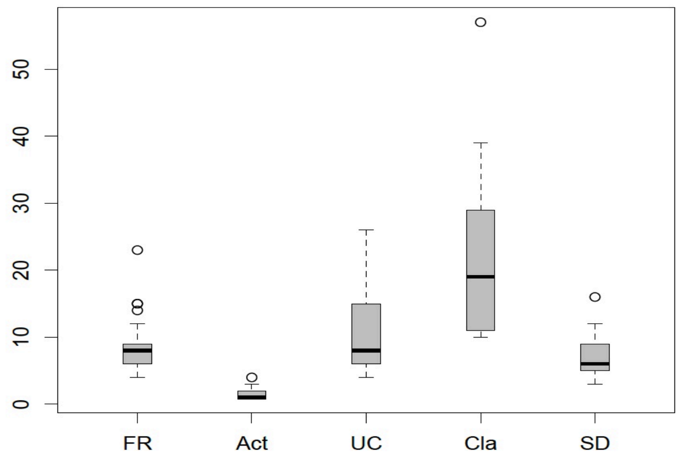
Fig. 2. The class diagram of the Archeotur project.

TABLE III
DESCRIPTIVE STATISTICS OF THE DEPENDENT AND INDEPENDENT VARIABLES CONSIDERED FOR THE ANALYSIS

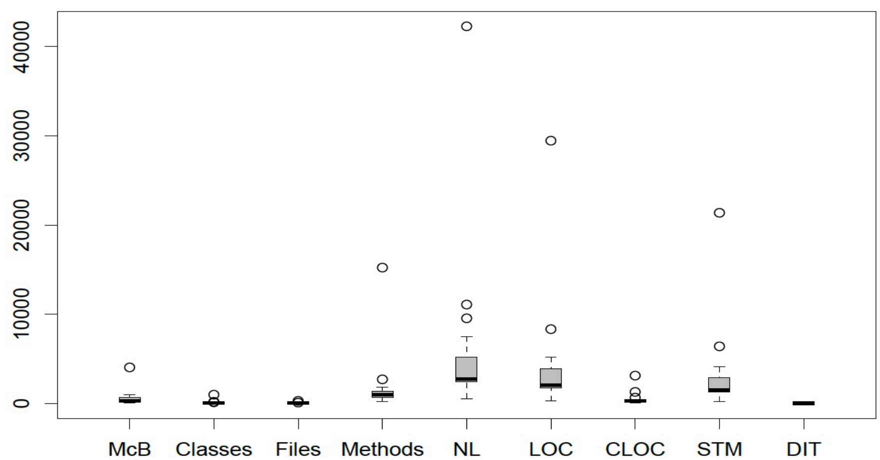
Variable	Min	Max	Mean	Median	Standard Deviation
FR	4	23	8.48	8	4.29
Act	1	4	1.59	1	0.8
UC	4	26	10.78	8	5.8
Cla	10	57	21.78	19	12.1
SD	3	16	7.07	6	3.025
McB	48	4030	517.52	282	747.91
Classes	12	967	89.22	54	178.62
Files	5	273	34.19	23	50.16
Methods	192	15222	1510.07	943	2795.71
NL	534	42287	5134.56	2740	7854.72
LOC	258	29456	3599.93	2037	5455.17
CLOC	12	3108	393.56	258	591.7
STM	163	21369	2714.44	1464	3969.11
DIT	2	4	2.44	2	0.58
Effort	30	113	58.82	55	21.04

the amount variable y changes when variables x_i changes 1 unit, and c is the intercept.

SWLR allows computing an equation in stages in which the choice of the independent variables is carried out by an automatic procedure. These variables can be chosen applying three approaches: forward, backward, or a combination of both [23]. The *forward* approach starts with no variables in the model. It tries out the variables one by one and includes them in the model if they are statistically significantly correlated with the dependent variable. The *backward* approach starts with all the variables and test them one by one. We remove the variables that are not statistically significant correlated with the dependent variable. We used here a combination of forward and backward approaches.



(a) RAD measures.



(b) SC measures.

Fig. 3. The boxplots of (a) RAD measure values, (b) and SC measure values.

At each step, this combined approach includes or removes variables one by one if they are or not statistically significant correlated with the dependent variable. We employed SWLR because this technique allows computing linear regression in stages and because it is widely used in the context of software prediction with appreciable results [17], [24], [25], [26].

To evaluate the goodness of fit of a model, we exploited the square of the linear correlation coefficient (i.e., R^2). This coefficient shows the amount of variance of the dependent variable explained by the model related to an independent variable. A good model should be characterized by a high R^2 value. We also considered the F value indicators and the corresponding p-value (denoted by Sign. F), whose high and low values, respectively, denote a high degree of confidence for the prediction.

C. The obtained effort prediction model

Before applying SWLR, we verified the following assumptions: *(linearity)* the existence of a linear relationship between independent and dependent variables; *(homoscedasticity)* the constant variance of error terms for all the values of independent variables; and *(normality)* the normal distribution of the error terms. We performed a log transformation of the input variables because the RAD measures were not normally distributed according to the results of a Shapiro test [27]. Furthermore, we performed the analysis of outliers, exploiting the Cook's distance and performed a stability analysis to eliminate influential observations [28].

The results of performed SWLR are summarized in Table IV. We can observe that the model built by RAD measures are characterized by a Sig. F value less than 0.05, thus the resulted model is significant. However, the obtained R^2 and F values are not so high.

In particular, the results revealed that best effort predictors include Cla, the number of classes in the RAD, and Act, the

TABLE IV
RESULTS OF SWLR FOR EACH DEPENDENT VARIABLE USING THE RAD MEASURES

Dependent variable	Independent variables	R^2	F	Sign. F (p-value)
Effort	Act	0.233	3.65	0.041
	Cla			
	Intercept			

TABLE V

EFFORT PREDICTED

ID	Number of participants	Act	Cla	Effort	Effort for participant
P1	4	2	31	56.53	14.13
P2	4	2	9	43.01	10.75
P3	3	1	7	49.13	16.38
P4	4	1	45	74.13	18.53
P5	4	2	22	52.40	13.10
P6	3	3	23	47.39	15.80

TABLE VI

REAL MEASURES AND EFFORT RESULTS

ID	Number of participants	Act	Cla	Effort	Effort for participant	Real effort for participant
P1	4	2	31	56.53	14.13	11.25
P2	4	2	35	58.06	14.52	24.75
P3	3	1	7	49.13	16.38	13.67
P4	4	1	30	67.77	16.94	17.25
P5	4	2	44	61.080	15.27	15.00
P6	3	3	23	47.39	15.80	16.33

number of use case actors, enabling the instantiation of the equation as follows:

$$Ln(\text{Effort}) = -0.272 * Ln(Act) + 0.221 * Ln(Cla) + 3.465$$

The final estimation model, when transformed back to the raw data scale, gives the following equation:

$$\text{Effort} = Act^{-0.272} * Cla^{0.221} * 31.96$$

A plausible justification for this outcome is that the number of classes in a RAD represents the basis for the next phases of the development process. That is, a developer uses these classes as starting point for implementation. Therefore, it seems reasonable that Cla provides useful information for an accurate prediction of the effort to implement apps. Furthermore, the number of use case actors give an indication of those interacting with the app, which is a crucial aspect for this kind of software.

V. MODEL APPLICATION

The successive edition of the MAD course was conducted with the same approach. Participants were 22. They were grouped in six teams. The data from the RAD have been collected as described in Section IV-A. We examined them and on the basis of the effort prediction measures, we proposed some little adjustments to the requirements concerning functionalities to be implemented as new classes. The number of participants for each team is reported in Table V, together with the number of actors and classes taken from the RAD and the estimated effort required for the project and for each participant. Let us assume that students work consecutively for three hours. This assumption is due to the fact that students follow also other courses and have only the afternoon free. Thus, the number of days to dedicate to the project implementation is 16. From these results it is possible to see that P1, P3 and P6 effort predictions are near to 16 days and we decided to leave them as they are; P2 and P5 have to be augmented to reach the required effort, while P4 has to be reduced.

The new versions of the RAD of these 4 projects have been analyzed and the results are in Table VI, together with the real effort. As it is possible to note, projects P4 and P5 reached a real effort near 16, while P2 dramatically exceed the estimation. Except P2, the other 5 projects respected the established time constraints.

VI. CONCLUSION AND DISCUSSION

In this paper we proposed to apply software estimation methods to the assignment of project works in mobile app development learning activities, specifically for Android application using a back-end server. The model has been created considering the data of the projects of year 2015 and has been applied to the RAD measures of the projects of year 2016. On the base of these measures, some adjustment of the project requirements have been done. The final estimation results on six projects were good for five of them, except for project P2, which doubled the required time to accomplish it. To better understand the reason why P2 was out of time we examined the type of the various applications. While P2 was a game with synchronization features, the others were dynamic web applications with Android interface and access to native features, e.g., an app which provides location-based search on swimming pools and events related to sea competitions. The servers-side of P2 was more complicated because of the need of connecting two players together in a single game session and transfer data messages between connected players. Thus, the metric values reported in this paper seems not to be effective for application with synchronization needs, considering the background of the students.

This approach may be useful to a teacher when it conducts the same type of course over the years. May be that when the project dataset grows collecting the data from different years the prevision may become more accurate. These considerations have to be confirmed by further work performed by collecting data of the successive MAD courses. In addition, a much deeper analysis and setting of groups' composition should also be part of a larger, extended and deeper experiment which may consider also different variables, such as the composition of programmers' teams, gender, previous background and motivation. These could be other factors affecting final results which can be examined in future investigations.

ACKNOWLEDGMENT

We thank all the students to the two courses and the IT managers, who judged the apps these students developed.

REFERENCES

[1] K. Molokken and M. Jorgensen, "A Review of Software Surveys on Software Effort Estimation," in *Proceedings of the International Symposium on Empirical Software Engineering (ISESE)*, pp. 223–230, 2003.

[2] R. Francese, C. Gravino, M. Risi, G. Scanniello, and G. Tortora, "On the Use of Requirements Measures to Predict Software Project and Product Measures in the Context of Android Mobile Apps: A Preliminary Study," in *Proceedings of the 41st Euromicro Conference on Software Engineering and Advanced Applications (EUROMICRO-SEAA)*, pp. 357–364, 2015.

[3] P. C. Blumenfeld, E. Soloway, R. W. Marx, J. S. Krajcik, M. Guzdial, and A. Palincsar, "Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning," *Educational psychologist*, vol. 26, no. 3-4, pp. 369–398, 1991.

[4] J. R. Savery and T. M. Duffy, "Problem Based Learning: An Instructional Model and its Constructivist Framework," *Educational technology*, vol. 35, no. 5, pp. 31–38, 1995.

[5] L. K. Michaelsen, A. B. Knight, and L. D. Fink, *Team-Based Learning: A Transformative Use of Small Groups*. Greenwood publishing group, 2002.

[6] L. Vygotsky, *Thought and Language*. MIT Press, 2012.

[7] M. Bieliková, "Experiences with Designing a Team Project Module for Teaching Teamwork to Students," *Journal of computing and information technology*, vol. 13, no. 1, pp. 1–10, 2005.

[8] W. Humphrey and J. Over, *Leadership, teamwork, and trust: Building a competitive software capability*. Addison-Wesley Professional, 2010.

[9] M. E. Joorabchi, A. Mesbah, and P. Kruchten, "Real Challenges in Mobile App Development," in *Proceedings ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 15–24, ACM Press, 2013.

[10] S. Kizaki, Y. Tahara, and A. Ohsuga, "Software Development PBL Focusing on Communication Using Scrum," in *Proceedeings of the IIAI 3rd International Conference on Advanced Applied Informatics (IIAIAAI)*, pp. 662–669, 2014.

[11] B. Silva and P. Girão, "A Study and a Proposal of a Collaborative and Competitive Learning Methodology," in *Proceedings of the IEEE Education Engineering (EDUCON)*, pp. 1011–1018, 2010.

[12] L. M. Regueras, E. Verdú, M. J. Verdú, and J. P. De Castro, "Design of a Competitive and Collaborative Learning Strategy in a Communication Networks Course," *IEEE Transactions on Education*, vol. 54, no. 2, pp. 302–307, 2011.

[13] J. C. Burguillo, "Using Game Theory and Competition-Based Learning to Stimulate Student Motivation and Performance," *Computers & Education*, vol. 55, no. 2, pp. 566–575, 2010.

[14] S. Yadav and J. Xiahou, "Integrated Project Based Learning in Software Engineering Education," in *Proceedings of the International Conference on Educational and Network Technology (ICENT)*, pp. 34–36, 2010.

[15] R. Francese, C. Gravino, M. Risi, G. Scanniello, and G. Tortora, "Using Project-Based-Learning in a Mobile Application Development Course - An Experience Report," *Journal of Visual Language and Computing*, vol. 31, pp. 196–205, 2015.

[16] L. Briand and I. Wieczorek, *Resource Estimation in Software Engineering*. Encyclopedia of Software Engineering, John Wiley & Sons, Inc. All, 2002.

[17] B. A. Kitchenham, E. Mendes, and G. H. Travassos, "Cross versus Within-Company Cost Estimation Studies: A Systematic Review," *IEEE Transaction on Software Engineering*, vol. 33, no. 5, pp. 316–329, 2007.

[18] M. J. Shepperd and G. F. Kadoda, "Comparing Software Prediction Techniques Using Simulation," *IEEE Transaction on Software Engineering*, vol. 27, no. 11, pp. 1014–1022, 2001.

[19] E. Mendes and N. Mosley, "Bayesian Network Models for Web Effort Prediction: A Comparative Study," *IEEE Transaction on Software Engineering*, vol. 34, no. 6, pp. 723–737, 2008.

[20] J. Ceddia and M. Dick, "Automating the Estimation of Project Size from Software Design Tools using Modified Function Points," in *Proceedings of the Sixth Australasian Conference on Computing Education*, vol. 30, pp. 33–39, Australian Computer Society, Inc., 2004.

[21] B. Bruegge and A. H. Dutoit, *Object-Oriented Software Engineering: Using UML, Patterns and Java, 2nd edition*. Prentice-Hall, 2003.

[22] C. Harpham and R. L. Wilby, "Multi-site Downscaling of Heavy Daily Precipitation Occurrence and Amounts," *Journal of Hydrology*, vol. 312, no. 1, pp. 235–255, 2005.

[23] T. J. Hastie and D. Pregibon, *Generalized Linear Models*. Wadsworth and Brooks/Cole, 1992.

[24] V. R. Basili, L. C. Briand, and W. L. Melo, "A Validation of Object-Oriented Design Metrics as Quality Indicators," *IEEE Transaction on Software Engineering*, vol. 22, no. 10, pp. 751–761, 1996.

[25] T. Gyimóthy, R. Ferenc, and I. Siket, "Empirical Validation of Object-Oriented Metrics on Open Source Software for Fault Prediction," *IEEE Transaction on Software Engineering*, vol. 31, no. 10, pp. 897–910, 2005.

[26] G. Scanniello, C. Gravino, A. Marcus, and T. Menzies, "Class Level Fault Prediction Using Software Clustering," in *Proceedings of the International Conference on Automated Software Engineering (ASE)*, pp. 640–645, IEEE Computer Society, 2013.

[27] S. S. Shapiro and M. B. Wilk, "An Analysis of Variance Test for Normality," *Biometrika*, vol. 3, no. 52, 1965.

[28] E. Mendes and B. Kitchenham, "Further Comparison of Cross-company and Within-company Effort Estimation Models for Web Applications," in *Proceedings of the International Software Metrics Symposium*, pp. 348–357, IEEE press, 2004.

Towards Formal Multimodal Analysis of Emotions for Affective Computing

Mehdi Ghayoumi
Artificial Intelligence Lab
Department of Computer Science
Kent State University
OH, USA
mghayoum@kent.edu

Maha Thafa
Artificial Intelligence Lab
Department of Computer Science
Kent State University
OH, USA
mthafar@kent.edu

Arvind K. Bansal
Artificial Intelligence Lab
Department of Computer Science
Kent State University
OH, USA
akbansal@kent.edu

***Abstract*—Social robotics is related to the robotic systems and human interaction. Social robots have applications in elderly care, health care, home care, customer service and reception in industrial settings. Human-Robot Interaction (HRI) requires better understanding of human emotion. There are few multimodal fusion systems that integrate limited amount of facial expression, speech and gesture analysis. In this paper, we describe the implementation of a semantic algebra based formal model that integrates six basic facial expressions, speech phrases and gesture trajectories. The system is capable of real-time interaction. We used the decision level fusion approach for integration and the prototype system has been implemented using Matlab.**

Keywords- Affective computing, Emotion recognition, Human-machine interaction, Multimedia, Multimodal, Decision level fusion, Social robotics.

I. INTRODUCTION

In the Human Compute Interaction (HCI) researches and studies, facial expression, speech and body movements have the major roles [8, 18 and 24]. Due to the aging society and increasing cost of health care, elderly care and assisted living, there has been significant interest in the development of social robotic systems that can interact with humans through the use of sensors. The social-robotic system can be humanoid, computers or intelligent machines as in Internet of Things who will interact with humans in the daily life. Interacting with humans requires understanding emotions [6] and emotions are based on a person's state of mind and partially regulated by personality, context and conditioning. Emotion is a language for communicating by feelings and it includes approval and disapproval to robotic systems. Interactive emotions [8, 16, and 18] are a subclass of human emotion analysis that humans use to interact with each other in close proximity. There are many interactive emotions that a person can express to machine during interaction, such as happiness, anger, embarrassment, surprise, rage, disappointment, confusion, elation, depression, approval and disapproval. Interactive emotions are expressed using a combination of verbal and nonverbal modes such as facial expressions [7, 10,], speech [17] including silence, gesture including body-posture and body-motion. Single mode may not give the emotion completely, or may be unavailable during emotive interaction. For example, the face may be occluded by the hands during sadness when a person is in deep pain or is crying. A person in shock or deep pain may not utter a single word. In the early stages of social robotics, most of the human-computer interaction in the service industry will involve brief commands or query by the human, and the robots will play a subordinate role rather than as companion role. Most of the emotion recognition will be limited to the integration of:

1) Facial expressions,
2) The limited amount of speech commands and emotional phrases to provide as an approval, disapproval, encouragement of a robot response or action, and
3) Gesture analysis of the upper body part involving head, hand, fingers, eyes, and lips.

The speech commands may be restricted to commands like "yes", "no", "don't like", "very good", "I am happy" etc. Some of these commands may have limited speech attributes such loudness showing disapproval or anger. Speech analysis can be done by a combination of text-to-speech conversion to understand the emotional phrases, and fast Fourier transform can be used to derive the variation of speech features such as energy, amplitude envelope, pitch during emotional variation. Real-time facial expression analysis in a video analysis where emotions and emotion transition can be studied by frame-to-frame analysis of facial expressions. Gesture analysis is done by analyzing video analysis and depth analysis as in Kinect based systems.

Currently, computational systems are limited to analyzing a single mode of emotion expression such as facial expression, speech, and (to some limited extent) multimodal analysis [22].

DOI reference number: 10.18293/DMS2016-030

The current integration of multimodal analysis systems of interactive emotions, lack:

1) A formal model to combine multiple modes such as facial expressions, speech analysis and gestures,
2) Complete catalog of upper body gestures, and
3) Capability of real-time analysis of facial-expressions and gestures.

In biometric systems, multimodal systems have some advantages which make the system accuracy and performance higher. Here we are using this model to achieve better results [23]. This research effort is in the direction of real-time integration of multimodal analysis system to derive emotion during HRI. A fusion module combines the weighted scores derived from each mode to derive the best emotion. The major contribution, here are:

1) Implementation of a real-time facial expression system based upon integration of geometric features, facial action units and facial symmetry [27],
2) Gesture recognition systems using fuzzy values,
3) Emotional Phrase lookup module,
4) Weighted score based Integration of a multimodal system based upon an abstract model of multimodal emotion analysis.

The rest of the paper is organized as follows. Section 2 describes background. Section 3 describes the overall architecture. Section 4 explains the speech recognition system and facial expression analysis will be explained in the section 5. Section 6 demonstrates the gesture modeling. Section 6 illustrates the implementation and performance results. Section 7 demonstrates the related works and the last section concludes the work, and describes the future directions.

II. BACKGROUND

This section describes the background material related to facial expression, speech analysis and gesture recognition and describe basic mathematical concepts needed for abstract modeling of the emotions.

A. *Components of Emotion Recognition*

There are three popular psychological theories of emotions: *James-Lange theory* [19], *Cannon-bard theory* [20] and *Schater-singer* theory [21]. *James-Lange* theory states that the mental state in response to the reactions which caused by external stimuli is emotion. *Cannon-Bard* theory is based upon anticipation rather than as a reaction to specific action. *Schachter-Singer* theory states that encountering an emotion requires both an interpretation of the bodily response as well as specific circumstance at a specific moment.

Also, there are three major classes of emotions:

1) Basic emotions,

2) Emotions that having same basic class, but having different intensity,

3) Mixed emotions that are a combination of one or more basic and/or mixed emotions.

Although, there are some disagreements among researchers, and a popular computational theory of Ekman [22] identifies six basic emotions: *happiness, sadness, surprise, disgust, anger* and *fear*. An example set of emotions having same basic class, but different intensities is {relaxed, happy, delighted, and euphoric}. Another set is {upset, anger, rage} etc. An example of mixed emotion is {amazed} that is a combination of {surprise and happiness} or {envy} which is the combination of {sadness and anger} or {despair} which is the combination of {fear and sadness}. In general, Facial Expressions have been done using these types of systems:

1) Facial Action Coding System (FACS) based on the simulation of facial muscle movement,
2) Geometric Features Modeling (GFM) based upon the movement of major feature-points of the face such as dynamic change in location endpoints and curvature of the mouth, eye, lips, forehead furrows and space between eyebrows.

The unit of FACS is an Action Unit (AU) that involves a segment of a muscle in facial expression. There are 17 major AUs involved in basic facial expressions. Examples of AUs involved in facial expressions are: inner brow raiser, outer brow raiser, brow lowered and drawn together, upper eye-lid raised, cheek raised, upper lip raised, lip corners pulled down, etc. The major geometric feature points, involved in facial expression analysis are given in Figure 1 which these features-points include:

1) 3 eyebrow points in each of the eyebrows:
$$b_1^L,\ b_2^L,\ b_3^L,\ b_1^R,\ b_2^R,\ b_3^R$$
2) 2 endpoints of eyes in each of the eyes:
$$e_1^L,\ e_2^L,\ e_1^R,\ e_2^R$$
3) Middle points eye-lid in each of the eyes:
$$el_L \text{ and } el_R$$
4) 2 endpoints of nose:
$$n^T \text{ and } n^B$$
5) 2 endpoints of mouth:
$$m^L \text{ and } m^R$$
6) 2 middle points of the mouth based on top and bottom lips:
$$m^T \text{ and } m^B$$
7) Chin-point denoted as:
$$ch\ .$$

The points shaded in dark black- e_1^L , e_2^L , e_1^R , e_2^R , n^T and n^B do not move, and act as reference-points. Remaining spotted-points move with emotions, and their displacement is used to derive the facial expression.

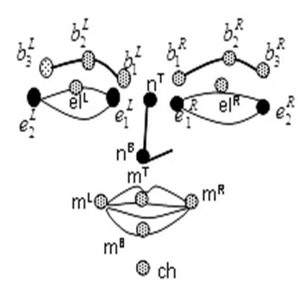


Figure 1. Major feature-points on the face

Emotional speech has multiple features such as phonemes, emotional phrases, amplitude, syllable envelope, pitch, rhythm, quantile and silence. Phonemes are the basic units of speech. During emotional interaction, pitch, amplitude, syllable envelope, duration of silence and utterances change significantly, act as parameters for the recognition of interactive emotions. Gesture is a nonverbal communication using perceptible bodily actions such as body-postures and body-part movements, including movements of the head, torso, hands, face and eyes. Different components of the emotions are measured using different sensors. Facial-Expression uses image analysis techniques to identify the movement of facial feature points, speech analysis uses wavelet analysis, FFT analysis, morphology analysis, text-to-speech conversion for phoneme detection and dictionary lookup to identify phrases. Gesture recognition requires image analysis to derive postures and video-frame analysis to derive motion of various body parts such as head, arm, eyes, hand, palm, fingers. The posture and motion are modeled as fuzzy values to reduce the computational space. The motion of the body parts can also be derived using skeletal and depth analysis used in Kinect.

B. Mathematical concepts

The Fuzzy values map a large value-space to a smaller finite space. The major advantages of the use of fuzzy values are:

1) Reduction of the computational complexity
2) Nearness to human perception and
3) Tolerance from the sensor noise.

We use two types of fuzzy sets:

1) Discrete fuzzy set, and
2) Ordered fuzzy sets.

A discrete fuzzy set has values that have no relationship that shows transitivity. For example, a head posture can be {rotated-left, rotated-right, normal, tilted-left, tilted-right, looking-down, looking-up}. An ordered fuzzy set shows transitive relationship between the values, and is used to model motion intensity in gesture analysis for better classification of emotion. For example, the speed of a head-motion can be modeled as {still, slow, normal, fast, very fast}. The values in the fuzzy set can be mapped onto the ordinals 0… 4:

Still →0, Slow→1, Normal→2, Fast→3 and Very Fast→ 4 (1)

The use of this mapping allows the use of comparison operators on ordered fuzzy sets. Cartesian product of the N sets returns a set of N-tuples such i^{th}-field of an element is a member of the i^{th} set as shown:

$$X_1 \times \ldots \times X_n = \{(x_1, \ldots, x_n) \mid x_i \in X_i \forall i = 1, \ldots, n\} \quad (2)$$

Two domains can be joined using:

1) Product-domain that uses the Cartesian product $A \times B$, or

2) A sum - domain that uses disjoint-union $A + B$, or

3) *Function Domain* mapping on lifted domains f: A⊥ B. Where ⊥ is the bottom symbol used to catch all ill-defined mappings.

III. Overall Architecture

Overall architecture (see Figure 2) has six major modules:

Unit 1 - Facial Expressions Subunit (FE)

The subunit takes a video-clip that is a sequence of frames, and extends the integration of FACS + geometric feature analysis technique for real-time basic face-expression recognition to derive the ranked subset of facial expressions for each frame in the video-clip. The analysis of a frame may result in more than one facial expression due to the:

1) Partial or full occlusion of the face due to gesture or head rotation,
2) Transition of emotions,
3) Inherent accuracies in the facial expression technique,
4) A variation of the facial expression due to the situation, personality or culture,
5) Low emotional intensity.

The outcome of this facial-expression analysis gives a sequence of subsets of derived possible emotions with the matching score of the form:

$$\langle FE_1, \ldots, FE_j, FE_{j+1}, \ldots, FE_N \rangle \quad (3)$$

Where N is the number of frames in the clip, FE_i is a subset of rank facial expressions of the form

$$\{(e_1, s_1) \ldots (e_m, s_k)\} \text{ For } k \geq 1, e_i \in \Sigma, s_i > \text{threshold and } s_i > s_{(i+1)} \quad (4)$$

Which e and s are the emotion and its corresponding score respectively.

Unit 2 Gesture Fuzzy Parameterization Module (GFP)

The model measures different postures and motion intensity and frequency of different emotional gesture patterns.

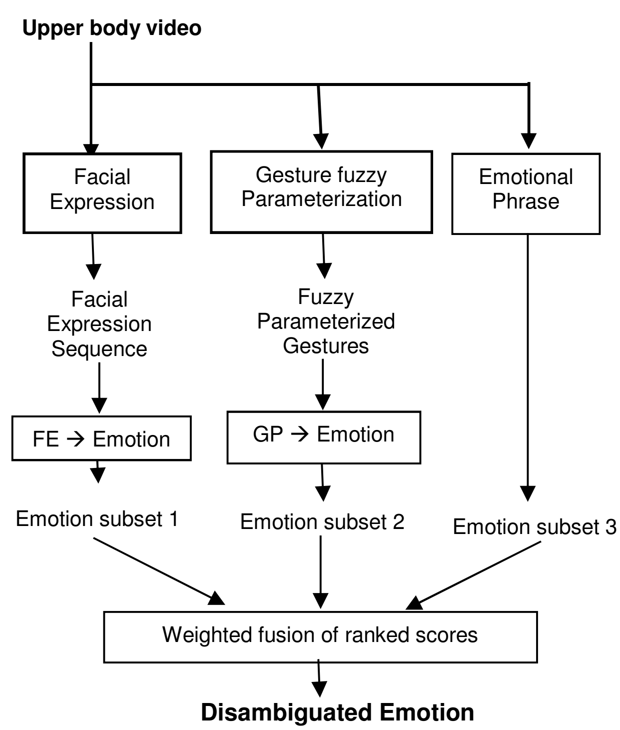


Figure 2. Overall architecture of multimodal fusion

The different postures are: body-posture, head-posture, shoulder-posture, hand-postures, palm-posture, finger-postures and eye-gaze and the various-motions are: head-motion, arm-motions, eye-motions, finger-motions. The details of the fuzzy parameterization and emotional head-motion gesture is given in Section 5.

Unit 3 - Emotional Phrases Module

Emotional phrase modules use a hash function to generate the index, and stores multiple emotions with a fuzzy intensity value. Once a phrase is recorded, then text-to-speech converter is used to derive the text. Individual words are looked up in the user-specified dictionary to remove the noise in text-to-speech conversion. The speech analysis system is used to derive the relative energy level. The energy level is parameterized to a fuzzy value, and the hash function is used in the derived text to identify the index of the speech. Using this index, the corresponding set of emotions that closely matches the intensity levels are derived.

Unit 4- FE (Emotion Module)

Many adjacent frames will have the same subset of facial expressions until the facial expression changes. We call this frame as the emotion transition point. Thus:

$$\langle FE_1, \ldots, FE_j, FE_{j+1}, \ldots, FE_N \rangle \rightarrow \langle (E_1, d_1), (E_2, d_2) \ldots, (E_M, d_M) \rangle \qquad (5)$$

Where E_I ($1 < I < M$) is a subset of Σ, and d_I is of the form start-frame: end-frame. The term d_I is used:

1) To reconcile the emotion in the fusion-module, and
2) To derive the duration of emotion to match with the duration of emotion derived from gesture analysis and emotional phrase analysis.

Unit 5- GFP (Emotion Module)

This module takes the fuzzy parameterized values of different body parts and their motion, concatenates them into a long string, and creates a string, and performs a similarity-based search in K-dimensional space where K is the number of fuzzy-valued component to identify a possible set of emotions. The attributes of fuzzy-vector representation of head-motion trajectory is hashed to derive the possible set of emotions based upon gesture. The duration of the body-motion is noted like unit 4, and its output is also on the form:

$$\langle (E_1, d_1), (E_2, d_2) \ldots, (E_M, d_M) \rangle \qquad (6)$$

Where E_I ($1 < I < M$) is a subset of Σ, and d_I is of the form start-frame: end-frame.

Unit 6 - Weighted Fusion of Emotion Module

The role of the weighted fusion module is to:

1) Fuse the information of the ranked emotions from the three modules to reduce ambiguity and improve ranking scores,
2) Derive the duration of emotion.

The input of the FE (emotion module (unit 4), and GP (emotion module (unit 5) and unit 3 are a sequence of set of rank emotion with the duration. Under the assumption, that emotions are expressed involuntarily in the facial expressions first, the start frame of the facial expression should occur before followed by the emotions derived from other two modules. To handle the issues that emotions may not be expressed by one or more modules, the weight is dependent upon:

1) Availability of the emotions from the specific mode,
2) The noise level.

For example, initial weight is w_1, w_2 and w_3 for the fusion of the corresponding modes, the weights w_k ($1 \leq i \leq 3$) is altered by:

$$\left(\sum_{i=1}^{i=3} w_i \Big/ \sum_{i \neq j, i=1}^{i=3} w_i \right) \qquad (7)$$

Since one of the modes is missing. The fusion is performed by:

1) Multiplying the ranked score of each emotion by the corresponding weight,
2) Adding the scores of the same derived emotions from different weights, and
3) Sorting the emotions by the descending order of the scores, and picking up the emotion with the highest score.

If the top two scores are very close, then it can be a case of emotional transition or mixed emotion.

IV. FACIAL EXPRESSION ANALYSIS

We extend the integration of FACS system interaction and geometric feature analysis [8] to make the facial expression analysis by using facial symmetry and invariance under head-motion. There are 13 moving-points (11 active points and 2 passive points) and 6 references-points. FACS system analysis has been used to derive the features-points that are significant during the expression of a specific facial expression. For example, for a surprise the all eyebrow points are uniformly raised; for happiness mouth corners are stretched, the eye-lid point gets lowered; for anger distance between eyebrows becomes smaller, inner eyebrow points get lowered. These FAUs have been translated to the corresponding feature-point movements as given in Table 1. We denote vertical-up motion by ↑, vertical-down motion by ↓, horizontally stretched outwards by '↔', horizontally compressed inwards by '↢', oblique-stretched downwards by '↘', oblique-stretched upwards by '↗'. If the emotion is symmetric, then the subscripts L and R have been omitted. If the movement is optional or shows higher intensity increase then it has been placed within the square brackets. Conjunction has been shown using concatenation. Essential feature-point have been within parenthesis () separated by ','. At least one of the essential feature point motion has to be present for the emotion to occur. Scores are associated with the presence of each feature-point motion.

TABLE1. Feature Point displacements

Facial Expressions	Major Feature-points displacements
Anger	$(e_1 \leftarrowtail el\uparrow) + [e_2\uparrow] + [m^T\uparrow m^B\uparrow]$
Disgusted	$(m^T\uparrow ch\uparrow) + [\{m^L, m^R\}\downarrow] + [m^B\uparrow]$
Fear	$(e_1\uparrow, m^L\downarrow m^R\downarrow) + [m^T\downarrow] + [e1\leftarrowtail]$
Happiness	$(m^L\nearrow m^R\nearrow, M^T\uparrow m^B\downarrow ch\downarrow m^L\longleftrightarrow m^R\longleftrightarrow)$
Sadness	$(el\downarrow m^L\longleftrightarrow m^R\longleftrightarrow) + [ch\downarrow]$
Surprised	$(e^1\uparrow e^2\uparrow e^3\uparrow el\uparrow ch\downarrow) + [m^T\uparrow m^B\downarrow]$

For each feature point, we measure the displacement distance and the direction of the displacement. Thus the derivable facial expressions are mapped to a vector of (displacement-distance ratio, direction).

Direction is a discrete-fuzzy set with six possible values:

1) Vertical-up,
2) Vertical-down,
3) Horizontal-compressed-inwards,
4) Horizontal-stretched-outwards,
5) Oblique-stretched-upwards, and
6) Oblique-stretched-downwards.

Facial expressions can be occluded due to:

1) Gestures such as hand covering face in case of sadness,
2) Lighting conditions, and
3) Head rotation or tilt.

In order to complete the information, we use the facial symmetry around nose to fill in the information about those facial expressions that show symmetry such as happiness, anger, surprise, sadness and fear. Disgust may shows asymmetric features. In order to variation of the displacement projection due to head motion, we use the distance-ratio (point-displacement from the relaxed state / distance between reference-points) which go with similar transformation.

For example, to keep the horizontal displacement we use distance between the two outer-eye corners: e_2^L and e_2^R in the denominator of the ratio; and for the two noses displacement. We use the distance between the two noses-points and n^B.

There are two types of motion-points:

A. Points that move in only in vertical up-down direction, such as:

$$e_1^L, e_2^L, e_1^R, e_2^R \text{ and}$$

B. Points that move in all four directions: vertical up-down and horizontal inside-outside motion such as:

$$el^L, el^R, m^L, m^R, m^T, m^B, ch$$

For the points that show motion in vertical up-down direction have only one entry in the *facial-expression ratio vector*, and points that show motion in up-down and stretch-compression mode have two entries in the vector. Based upon the emotion, some of the entries may not change during that emotion.

For example, in surprise, only, $el^R, e_2^L, e_3^L, e_2^R, e_3^R, m^T, m^B$ and ch change. This characteristic of facial-expression ratio-vector provides invariance against head-motion as well as specific characterization of facial-expressions.

V. Speech Analysis

Embedding the component of emotion processing into existing speech systems makes them more natural and effective. Several approaches to recognize emotions from speech have been reported. In a conversation, non-verbal communication carries an important information like the intention of the speaker. In addition to the message conveyed through text, the manner in which the words are spoken, conveys essential non-linguistic information. The same textual message would be conveyed with different semantics by incorporating appropriate emotions. Spoken text may have several interpretations, depending on how it is said. For example, the word 'OKAY' in English, is used to express respect, disbelief, agreement, and disinterest. Therefore, understanding the text alone is not sufficient to interpret the semantics of a spoken utterance. However, it is important that, speech systems should be able to process the non-linguistic information such as emotions, along with the message. Choosing suitable features for developing any of the speech systems is a crucial decision.

We have three important speech features, namely:

1) Excitation source,
2) Vocal tract system, and
3) Prosodic features.

VI. Gesture Analysis

Gesture parameterization has two modules:

a) Deriving the posture of upper-body parts and their motions,

b) Mapping fuzzy to actual values to reduce the search space.

These fuzzy values are concatenated so that all the values from discrete sets are concatenated together, and all the values from the ordered sets are grouped together. This separation is necessary because mismatch in discrete sets leads to failure, while mismatch in ordered sets is permissible. Abstract modeling of emotion requires functional mapping of Cartesian product of different components to derivable emotions. Since all the component tuples may not map to valid emotional elements in the emotional domain, we make the emotion domain a lifted domain by introducing a bottom symbol ⊥ in the set of well-defined emotions. The lifted domain allows for catching the error conditions when the tuple of fuzzy component values do not map to any specific emotion.

Fuzzy values are calculated using statistically derived thresholds. There are two types of sets:

1) Discrete sets where the values are not ordered and ordered sets,
2) Ordered sets are used in modeling the extent of posture variation and intensity of the motions of various gestures.

The parameters of the motion are:

a) Start-position,
b) End-position,
c) Frequency,
d) Speed,
e) Attack,
f) Relaxation.

The attack is the rate of change of speed until the motion attains the peak speed, relaxation is the rate of reduction of speed to the speed reduces from the peak speed to no motion. The mapping of the components is described by equation (7).

$$\text{Posture}_1 \times \ldots \times \text{Posture}_M \times \text{Motion}_1 \times \ldots \times \text{Motion}_M \rightarrow \text{set of possible emotions} \quad (7)$$

VII. Implementation and Performance Results

The implementation can be divided into 3 major steps as follows:

A. *Implementing Facial Expression Sequence Analysis*

We extract the main parts of the face such as eyes, eyebrow, nose and mouth, then find the key points in each segment as shown in the figure 1 then we extract the feature vectors from extracted key points and train the networks.

B. *Implementing Head Gesture Analysis*

Human head movement is very important in general conversation and communication. Despite the influential role of the head gestures, very little research has examined the gesture's role in the robot-human interaction process. In software module, the pose of the human head is estimated with a constraint that the human head is a 3 DOF rigid object which has yaw, pitch and roll movements. We have used geometric head pose estimation algorithm which estimates the head pose through a standard webcam of the computer. The details of this algorithm can be found in [12].

C. *Implementing Emotional Phrase Matching*

A cepstrum is obtained by computing the Fourier Transform of the logarithm of the spectrum of a signal. There are different kinds of cepstrum such as complex cepstrum, real cepstrum, phase cepstrum and power cepstrum. The power cepstrum is used in speech synthesis applications and here we use it. The approach based on decision-level fusion obtained. The performance of the classifier was 94.6%, both for the best probability and for the majority vote plus best probability approaches.

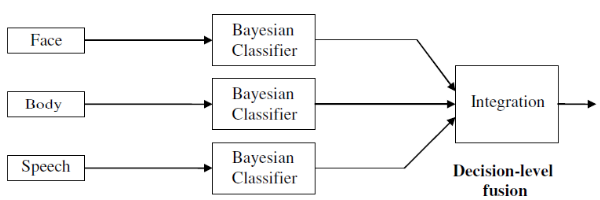


Figure 3. The decision level fusion

Table 2 shows the performance of the system with decision level integration using the best probability approach. Anger has the emotion recognized with highest accuracy.

TABLE 2. Decision level integration with the best probability approach

Anger	Happy	Sad	Surprise	Disgust	Fear	
98.3	0	0	0	4.3	0	**Anger**
0	**95.4**	0	7.2	0	0	**Happy**
3.1	0	**92.1**	0	2.7	2.2	**Sad**
9.3	10.4	0	**87.5**	2.3	5.8	**Surprise**
7.2	0	4.1	3.3	**90.2**	4.2	**Disgust**
0	0	0	12.2	11.1	**83.3**	**Fear**

VIII. Related Works

There are many related works in the facial expression analysis [1, 10, 25 and 26], gesture analysis [2, 4, 12, and 18] emotion recognition in speech [9] and multimodal fusion [3, 4]. Castellano et al. [7] extended their work to multimodal framework integrating face-expression, body-gestures, and speech. A Bayesian classifier was used for feature level fusion and decision level fusion. A comparison between unimodal, bimodal, and multimodal classification showed that multimodal classification is better. There is an additional need to identify features that are relevant to the dynamics of expressive emotions, however, their study is limited to identifying eight emotions [7]. We have been influenced by the research to derive from the research to analyze facial expressions based upon action units and map action units based movement to geometric feature-point movements [8]. The use of geometric feature-points and fusion allows for better accuracy in our research. In addition, we use abstract model for gesture analysis. Our geometric point movement based upon study of facial action units is generally enough to analyze finer classification of emotions and mixed emotions. Many interesting works about audio-visual fusion/mapping has been proposed for multimodal information processing. For instance, speech based facial animation [13], and audio-visual based emotion recognition [14]. There is also some work for head motions [15] and body gestures [16], however, most of them just focused on the gesture recognition.

IX. Conclusion and Future Works

In this paper, we have described a detailed methodology and an initial prototype implementation of real-time multimodal fusion to derive interactive emotion for interaction with social-robots and intelligent machines with limited emotional phrase based interaction. The proposed integrated system has many novelties such as: an abstract model of fusion based upon a semantic algebra that the maps Cartesian product of different components to derivable emotions, the use of invariant displacement of geometric feature-points to identify facial-expressions, and Gestures based upon head-trajectory and fuzzy values of other upper body parts to reduce the search space. Currently, the gesture based system is limited to, image analysis of feature-points in the head and hand to derive posture. We are looking into Kinect based analysis to integrate skeleton based body posture, motion and depth analysis [15] for better accuracy.

References

[1] R. Adolphs. "Recognizing emotion from facial expressions: psychological and neurological mechanisms," *Behav. Cogn. Neurosci. Rev*., Vol. 1, 2002, pp. 21-62.

[2] V. Bevilacqua, D. Barone, F. Cipriani, G. D'Onghia et al., "A new tool for gestural action recognition to support decisions in emotional framework", *Proceedings of the IEEE Symposium of Innovations in Intelligent Systems and Applications* (INISTA)*,* 2014, pp. 184-191.

[3] G. Caridakis, G. Castellano, L. Kessous, A. Raouzaiou, L. Malatesta, et al. "Multimodal emotion recognition from expressive faces, body gestures and speech"; *Artificial Intelligence and Innovations: From Theory to Applications*, Springer Berlin Heidelberg, 2007, pp. 375-388.

[4] G. Castellano, S. D. Villalba and A. Camurri. Recognizing human emotions from body movement and gesture dynamics"; *Affective Computing and Intelligent Interaction*, Springer Berlin Heidelberg, pp. 71-82.

[5] P. Ekman and W. Friesen, "Facial Action Coding System: A Technique for the Measurement of Facial Movement," *Consulting Psychologists Press*, Palo Alto, 1978.

[6] J. M. Fellous and M.A.Arbib,"Who needs emotions? The brain meets the robots", Oxford press, 2005.

[7] L. Gang, L. Xiao-hua, Z. Ji-Liu and G. Xiao-gang, "Geometric feature based facial expression recognition using multiclass support vector machines," *IEEE International Conference on Granular Computing (GRC '09)*, 2009, pp. 318-321.

[8] M. Ghayoumi and A. K. Bansal, "Unifying Geometric Features and Facial Action Units for Improved Performance of Facial Expression Analysis," Proceedings *of the International Conference on Circuits, Systems, Signal Processing, Communications and Computers* (CSSCC 15), pp. 259-266.

[9] C. H. Lim, E. Vats and C. S. Chan, "Fuzzy human motion analysis: A review," *Pattern Recognition,* Vol. 48, 2015, pp. 1773-1796.

[10] S. Mitra and T. Acharya. "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 37, 2007, pp. 311-324.

[11] C. M. Lee, S. S. Narayanan, "Towards Detecting Emotions in Spoken Dialog," *IEEE Trans. On Speech and Audio Processing*, Vol. 13, No. 2, 2005, pp. 293-303.

[12]] M. Khan, S. Rehman, Z. Lu and H. Li, "Head Orientation Modeling: Geometric Head Pose Estimation using Monocular Camera," in Proceedings of the 1st IEEE/IIAE International Conference on Intelligent Systems and Image Processing, 2013.

[13] Alberto B, Piero C, Giuseppe RL, Giulio P Lucia" a new WebGL-based talking head," 15th Conference of the International Speech Communication Association, Singapore 2014.

[14] Cowie, R, Douglas-Cowie, E "Emotion recognition in human-computer interaction." IEEE Signal Processing Magazine. pp. 33-80, 2001.

[15] Bo X, Georgiou Panayiotis G, Brian Baucom, Shrikanth S Narayanan, "power-spectral analysis of head motion signal for behavioral modeling in human interaction," I.E. International Conference on Acoustics, Speech, and Signal Processing, 2014.

[16] Welbergen HV, Reidsma D, Ruttkay ZM, Zwiers EJ D, "A BML realizer for continuous, multimodal interaction with a virtual human." Multimodal User Interf 3D4:271-284, 2010.

[17] Oh-Wook Kwon, Kwokleung Chan, Jiucang Hao, Te-Won Lee, "Emotion Recognition by Speech Signals" - INTERSPEECH, 2003.

[18] M. Ghayoumi, A. Bansal, "An Integrated Approach for Efficient Analysis of Facial Expressions", SIGMAP 2014.

[19] A, Walter. "The James-Lange Theory of Emotions: A Critical Examination and an Alternative Theory." The American Journal of Psychology 39: 106-124.

[20] Friedman, B.H. "Feelings and the body: The Jamesian perspective on autonomic specificity of emotion." Biological Psychology 84: 383-393 (2010).

[21] Cotton, J. L. "A review of research on Schachter's theory of emotion and the misattribution of Arousal." European Journal of Social Psychology 11: 365-397 (1981).

[22] P. Ekman, Facial expression and emotion. American, 8 (4): 384-392, 199.

[23] M. Ghayoumi, "A Review of Multimodal Biometric Systems Fusion Methods and Its Applications.", ICIS, USA, 2015.

[24] H. Abrishami Moghaddam and M. Ghayoumi, "Facial Image Feature Extraction Using Support Vector Machines.", Proc. VISAPP, Setubal, Portugal, 2006.

[25] Ghayoumi, M., Bansal, A. K."Multimodal Architecture for Emotion in Robots Using Deep Learning." Future Technologies Conference, San Francisco, United States, FTC 2016.

[26] M. Ghayoumi, A. K. Bansal, "Emotion in Robots Using Convolutional Neural Networks.", ICSR 2016.

[27] M. Ghayoumi, A. K. Bansal, "Real Emotion Recognition Algorithm by Detecting Symmetry Patterns with Dihedral Group.", MCSI 2016.

Parameter Calibration Method of Microscopic Traffic Flow Simulation Models based on Orthogonal Genetic Algorithm

Yanfang Yang
School of Traffic and Transportation
Beijing Jiaotong University
Beijing, P. R. China
E-mail: yangyf@bjtu.edu.cn

Yong Qin
State Key Laboratory of Rail Traffic Control and Safety
Beijing Jiaotong University
Beijing, P. R. China
Corresponding author, Email: yqin@bjtu.edu.cn

Honghui Dong
School of Traffic and Transportation
Beijing Jiaotong University
Beijing, P. R. China
Email: hhdong@bjtu.edu.cn

Qing Zhang
Beijing Research Center of Urban Traffic Information Sensing and Service Technologies
Beijing Jiaotong University
Beijing, P. R. China
Email: 15125808@bjtu.edu.cn

***Abstract*—Traffic microscopic traffic simulation models have become extensively used in both transportation operations and management analyses, which are very useful in reflecting the dynamic nature of transportation system in a stochastic manner. As far as the microscopic traffic flow simulation users are concerned, the one of the major concerns would be the appropriate calibration of the simulation models. In this paper a parameter calibration method of microscopic traffic flow simulation models based on orthogonal genetic algorithm is presented. In order to improve the capacity of locating a possible solution in solution space, the proposed method incorporates the orthogonal experimental design method into the genetic algorithm. The proposed method is applied to an arterial section of Ronghua Road in Beijing. Through comparing with the parameter calibration method based on genetic algorithm, the advantage of the proposed method is shown.**

Keywords-Microscopic traffic flow simulation model; Parameter calibration; Orthogonal genetic algorithm; VISSIM

I. INTRODUCTION

Traffic simulation has become an important and popular tool in modeling transport system, with the progress of simulation technologies [1]. Traffic simulation models could be divided into three categories, including microscopic, macroscopic, mesoscopic simulation models. Microscopic simulation models simulate traffic at a level of individual vehicles [2]. Car-following and lane-changing models are the two fundamental components in microscopic simulation models. Macroscopic simulation models simulate transportation network section-by-section rather than tracking individual vehicles. Mesoscopic traffic simulation models combine the properties of the microscopic and macroscopic simulation models. For the traffic simulation models, the simulation results depend on the initial choice of the model [3] and the success of the calibration process [4].

The calibration of traffic microscopic simulation models is defined as the process of finding optimal parameters to match the field data so model will accurately represent field measure or observed traffic condition [5]. The optimization task involves comparing and minimization differences of selected indicators, e.g., travel time and queuing length [4], delays [6], travel time distribution [7], saturation flow rates [8] and emission [9], between the calibration model and the ones counted and measured in local traffic network.

Calibrating traffic model of bigger special and time scopes of a traffic network needs deal with a larger number of input parameters and calculating processes. In order to decrease time consumption, artificial intelligent techniques are applied into the calibration of traffic microscopic simulation model. Genetic algorithm (GA) has become the most common used calibration algorithm for input parameters of the simulations [5, 10-13], since Cheu et al., firstly used GA calibrating FRESIM model [14]. Other intelligent algorithms are also used in the calibration of traffic simulation, e.g., perturbation stochastic approximation (SPSA) scheme [15], particle swarm optimization (PSO) [16], and neural network approach [4]. These methods automate the calibration process to a certain degree and it was generally reported that they improve simulation performance over the default model parameter values.

Microscopic simulation is a complex system that all parameters work together to influence its modeling results. In calibrating such a complex model, users could get trapped in

DOI reference number: 10.18293/DMS2016-047

the local optima of the objective function, due to the high dimension and numerous local optima. This paper focuses on the above question of GA when calibrating the driving behavior model parameters in VISSIM.

The exposition of this paper is as follows: the next section implements the orthogonal genetic algorithm (OGA). Some studies have found that applying an experimental design method (orthogonal design) into GA may overcome the limitation mentioned above [17-18]. The third section gives the procedure of calibrating the microscopic traffic flow simulation model based on OGA. The fourth section applies OGA calibration method to the calibration of a signal intersection in Beijing, and comparing with the GA and orthogonal design method respectively. The final section summarizes the paper.

II. CALIBRATION METHOD BASED ON OGA

The proposed calibration method employs an orthogonal genetic algorithm. This section contains a brief overview of the VISSIM calibration parameters set, and the fundamentals of the OGA, including the structure of chromosome, fitness function and orthogonal crossover decoding.

A. Selection of Parameters

In this paper, a microscopic traffic flow simulation, VISSIM, is selected as the basic platform for the parameter calibration. VISSIM models the psychophysical driver behavior and attempts to capture both the physical and the human components of traffic [19]. Parameters of two driving behavior models are in considered in this paper: the car-following model and the lane-changing model. After parameter sensitivity analysis, i.e., one-way Analysis of Variance (ANOVA)[20], four parameters are selected as the calibration parameters. Table 1 lists the calibration parameter set, including default value, the minimum and maximum value.

TABLE 1 CALIBRATION PARAMETER SET

Parameters (x_j)	**Unit**	**Default Value**	**Min (u_j)**	**Max (v_j)**
x_1 Average standstill distance	m	2	0.5	3
x_2 Additive part of desired safety distance	NA	2	0.5	3
x_3 Multiple part of desired safety distance	NA	3	1	6
x_4 Maximum deceleration	m/s^2	4	2	6

B. The Structure of Chromosomes

Supposing P_i (x_{i1}, …, x_{iN}) is the ith chromosome, x_{ij} is the jth parameter value in the ith chromosome. $\forall$ i=1,2, …, M, j=1, 2,…, N=4. M is the total number of chromosomes and N is the number of parameters to be calibrated. [$\boldsymbol{l}$, $\boldsymbol{u}$]= [(l_1, …, l_N), (u_1, …, u_N)] defines the feasible solution space and the corresponding domain of x_j is [l_j, u_j] (e.g., the domain of x_1 is [0.5, 3] as listed in Table 1). In this paper, parameters to be calibrated are coded into chromosomes, which quantized by orthogonal design. We quantize the domain [l_j, u_j] of x_j into Q levels, where the design parameter Q is odd.

Algorithm 1 shows the procedure of constructing chromosome. Firstly, we calculate M, where M= Q^J, J is the smallest positive integer fulfilling $J \geq \frac{\log(N(Q-1)+1)}{\log Q}$. Secondly, we construct the orthogonal array $L_M(Q^N)$ corresponding to the chromosome. Each element a_{ij} of the orthogonal array $L_M(Q^N)$ represents the levels numbers in orthogonal design, $a_{ij} \in \{0,1,...,Q\text{-}1\}$, $\forall$ i=1,2, …, M, j=1, 2,…, N. Finally, the corresponding parameter value x_{ij} of a_{ij} in feasible solution pace [l_j, u_j] is calculated by the equation as shown in follows:

$$x_{ij} = l_j + a_{ij} \times \frac{u_j - l_j}{Q-1}, \quad \forall 0 \leq a_{ij} \leq Q-1 \qquad (1)$$

Algorithm 1: Constructing chromosomes

Step 1. Calculating the number of chromosome, M=Q^J. J is the smallest positive integer fulfilling $J \geq \frac{\log(N(Q-1)+1)}{\log Q}$.

Step 2. Construct the orthogonal array $L_M(Q^N)$

(1) Construct the basic columns：

for k=1 **to** J **do**

$j = \frac{Q^{k-1}-1}{Q-1}+1$

for i=1 **to** Q^J **do**

$a_{ij} = \left\lfloor \frac{i-1}{Q^{J-1}} \right\rfloor$ **mod** Q

end for

end for

(2) Construct the non-basic columns:

for k=2 **to** J **do**

$j = \frac{Q^{k-1}-1}{Q-1}+1$

for s=1 **to** j-1 **do**

for t=1 **to** Q-1 **do**

$a_{j+(s-1)(q-1)+t} = (a_s \times t + a_j)$ **mod** Q

end for

end for

end for

(3) Selecting the first N columns to construct the chromosome encoding array $L_M(Q^N)$

Step 3. Calculate x_{ij} to construct chromosome using (1).

C. Fitness Function

The fitness function is a combination of the root mean absolute square error (RMASE) of travel time (TT) and maximum queue length (MQL) and between the VISSIM output and field data. The fitness function takes the form of

$$F_{total} = \frac{1}{\xi \cdot E_1 + (1-\xi)E_2} \quad (2)$$

$$E_n = \sqrt{\frac{1}{T}\sum_t \left(\frac{d_{nt}^s - d_{nt}^o}{d_{nt}^o}\right)^2} \quad \forall \quad n = 1, 2, t = 1, 2, \ldots, T \quad (3)$$

where
E_1 = the RMASE of TT
E_2 = the RMASE of MQL
T= the number of detector station;
d_{nt}^s = the simulation model output of tth sensor
d_{nt}^o = the field data of tth sensor station.

D. Adaptive orthogonal crossover operator

For each pair of parents (denoted by $P_1(x_{11}, x_{12}, x_{13}, x_{14})$ and $P_2(x_{21}, x_{22}, x_{23}, x_{24})$), performing the adaptive orthogonal crossover operation with the probability of crossover p_c. Adaptive orthogonal crossover algorithm is shown in Algorithm 2. σ_0 is the similarity threshold of each dimension between two parents .

Algorithm 2: Constructing chromosomes

Step 1. Calculating the number of similar dimension using $b = Num(|x_{1,j} - x_{2,j}| > \sigma_0)$

Step 2. Constructing orthogonal array $L_E(F^b) = [a_{ts}]_{E\times b}$, with b factors and F levels, using Algorithm 1.

Step 3. Generating E chromosomes as the potential offspring, $P_t'(x_{t,1}', x_{t,2}', x_{t,3}', x_{t,4}')$, $\forall$ t=1,2,…,E.

for t=1 **to** E **do**
s=1
for j =1 **to** 4 **do**
if $|x_{1j} - x_{2j}| \le \sigma_0$

$$x_{tj}' = \frac{x_{1j} + x_{2j}}{2}$$

else if $|x_{1j} - x_{2j}| > \sigma_0$

$$x_{tj}' = \min(x_{1i}, x_{2i}) + a_{ts}\frac{\max(x_{1i}, x_{2i}) - \min(x_{1i}, x_{2i})}{F-1}$$

s= s+1
end if
end for
end for

Step 4. Calculating the fitness values corresponding to each potential offspring and selecting the chromosome corresponding to the max fitness value as the offspring of two parents P_1 and P_2.

In this paper， σ_0 =0.005 and F=2.

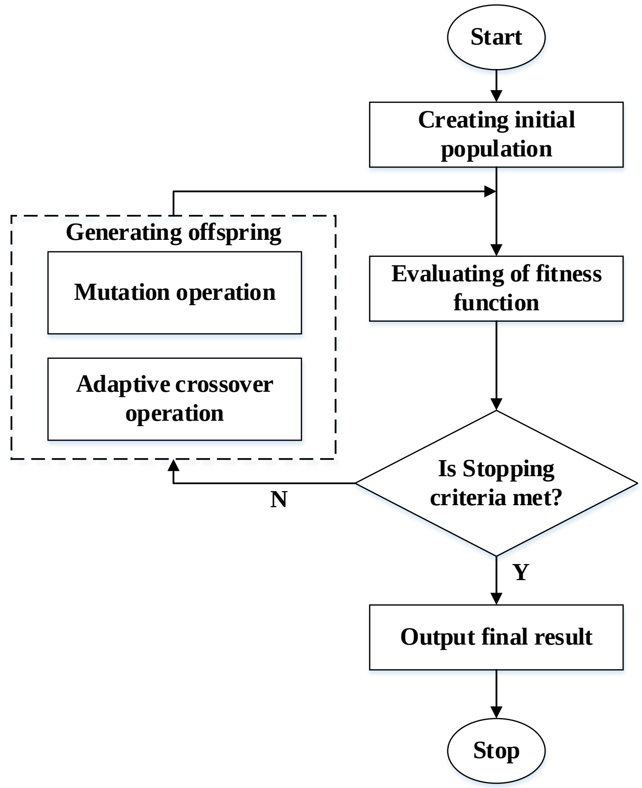


Figure 1 Calibration procedure

III. CALIBRATION PROCEDURE

We develop the parameter calibration program using VB language. The details of overall algorithm are as follows (see Fig. 1).

Step 1：Initialization

Executing Algorithm 1 to generate M potential parents (e.g., levels Q = 9 and $M = Q^J = 9^2 = 81$). The microscopic simulation traffic simulation model VISSIM is run with the M parameter group as the input file successively. Then M fitness values corresponding to the potential parents are calculated. Sort the potential parents in a descending sort order, according the fitness values. Select the first I := 50 parents as the potential initial population P_0. In order to keep the individual distributing uniformly, we select D := 10 chromosomes randomly from the D potential initial population as the initial population, denoted by $P_{gen=1}$.

Step 2：Adaptive crossover operation

For an arbitrary pair of parents, adaptive crossover operation is performed with the probability of crossover P_c (e.g., P_c =0.75) using algorithm in Algorithm 2 and generate offspring population C_{gen}.

Step 3：Mutation operation

Each chromosome in P_{gen} would undergo mutation operation with the probability of mutation p_m (e.g., p_m = 0.1). The mutation operation is as follows: (1) randomly generating an integer $j \in$ [1, N] and a real number $z \in [l_j, u_j]$; (2) replacing the *j*th component of the chosen chromosome by *z* to get a new chromosome. The mutation operation generates a new population denoted by G_{gen}. The fitness values corresponding to each new chromosome in G_{gen} are calculated by running VISSIM model.

Step 4: Selection operation

In order to maintain the population diversity, we sort the population (P_{gen} + C_{gen} + G_{gen}) in a descending sort order according the fitness values , then select the first $\lfloor D*70\% \rfloor$ chromosomes and randomly select $D-\lfloor D*70\% \rfloor$ chromosomes from the rest of (P_{gen} + C_{gen} + G_{gen}) to constructing the next population P_{gen+1}.

Step 5: Check stopping rules

Supposing the max fitness values of iteration gen is $F_{\max}^{gen}$. If gen = the maximum number of iterations or $|F_{\max}^{gen}-F_{\max}^{gen\text{-}1}|\leq 0.005$, the program stops. Otherwise, go to Step 2 and gen = gen+1.

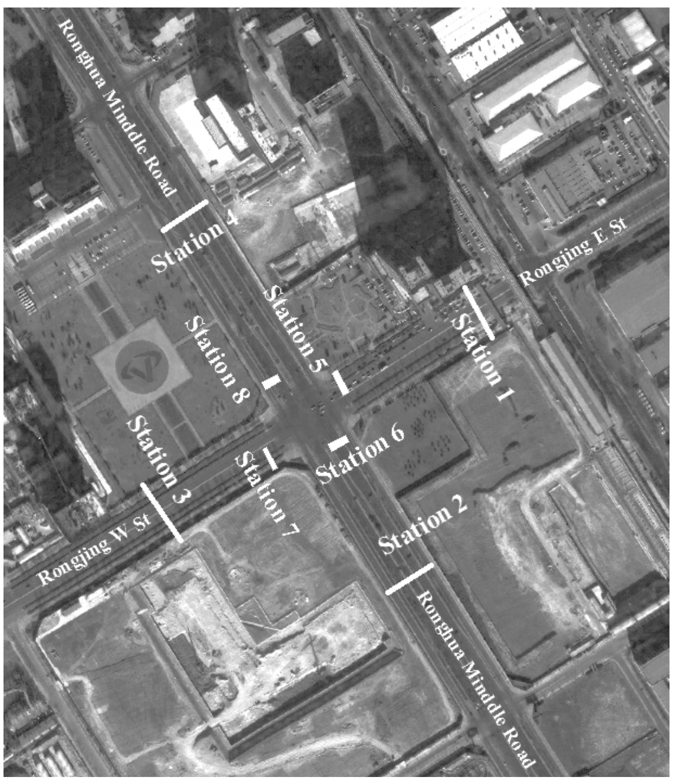


Figure 2 The location distribution of detector stations

IV. CASE STUDY

A signal intersection constructed by the Ronghua middle-road and the Rongjing road in Beijing is selected as the test bed. The intersection locates at the arterial section of Beijing economic-technological development area. The location and map are shown in Fig. 2. Field data were gathered at evening peak period (18:15-19:15) in July 25, 2011. The traffic volumes were video-taped by four cameras, which were located at the Station 5-Station 8 in Fig. 2, respectively. We recorded the maximum queue length at the entrance (i.e., Station 5-Station 8 in Fig. 2) into the intersection every signal cycle and use the mean of the above queue length as the hourly maximum queue length. The travel time was collected by floating cars. Fig. 2 shows the start and end points of travel time collection detectors, e.g., the Northwest-bound travel time collection is from Station 2 to Station 4, the Southeast-bound travel time collection is from Station 4 to Station 2.

The traffic model used is VISSIM Version 5.30. After building the VISSIM model, we apply the OGA to the parameter calibration. The default size of initial population *D* is 10. When the difference of maximum fitness values of two consecutive iterations is no more than 0.005 or the maximum number of iterations is 50, the program stops. Table 2 lists the relative errors of traffic volumes between VISSIM output results and field data at four entrances respectively, which shows that the values are all no more than 2.21%.

TABLE 2 VOLUME CALIBRATION OF THE INVESTIGATED INTERSECTION

	Simulation Results （vehicle/h）	**Field Data （vehicle/h）**	**Relative Error**
Northeast-bound	1173	1170	0.26%
Southwest-bound	486	497	2.21%
Southeast-bound	1766	1782	0.9%
Northwest-bound	525	528	0.57%

The proposed method is compared with calibration with the GA method and the orthogonal design method. In GA-based parameter calibration experiments, the fitness function, the size of initial population and the stopping rules are consistent with the OGA-based parameter calibration experiments. In the orthogonal design method, level number *Q* is 9, and the weight coefficient is 0.5. In the orthogonal design experiments, the parameter set is quantized by orthogonal design. The level number *Q* is the same with the OGA and the GA. The orthogonal array is constructed using Algorithm 1. Using the Q^J = 81 parameter groups as the VISSIM input data and selecting the maximum fitness value as the final output results.

We compare the maximum fitness values corresponding to four methods (i.e., default value, OGA, GA and the orthogonal method). The RMASE of travel time, RMASE of maximum queue length and fitness values are listed in Table 3. The maximum fitness value of OGA is 19.43, which is much bigger than that of other methods.

TABLE 3 COMPARATIVE ANALYSIS OF THE FITNESS FUNCTION VALUE

Methods	Parameter values	TT RMASE	MQL RMASE	Weight	Fitness
Default	[2,2,3,-4]	0.0607	0.3372	0.5	5.03
OGA	[1.125,0.5, 2.25,-2.75]	0.0225	0.0805	0.5	19.43
GA	[1.2,2.6, 1.33,-6]	0.0287	0.1380	0.5	11.99
Orthogonal method	[0.5,0.813, 1.625,-5]	0.0430	0.1378	0.5	11.06

Calculate the average errors for maximum queue length and travel time, respectively. The formula is as follows:

$$\varepsilon_i = 0.25\sum_{t=1}^{4}\frac{|d_{nt}^{s} - d_{nt}^{0}|}{d_{nt}^{0}} \quad n \in \{1,2\} \qquad (4)$$

where,

ε_1 = the average errors of TT

ε_2 = the average errors of MQL

d_{nt}^{s} =simulation output of *t*th entrance

d_{nt}^{0} = field data of *t*th entrance

t=1, 2, 3, 4 represent four entrance, i.e., northeast-bound, southwest-bound, southeast-bound and northwest-bound.

Fig. 3 and Fig. 4 represent the relative error of each access respectively. The relative error of the proposed method is less than that of other methods. Meanwhile, the relative error of travel time is higher than that of queue length, because investigators collect the queue length according to estimating the length of vehicles. It is obviously that the error must be existed.

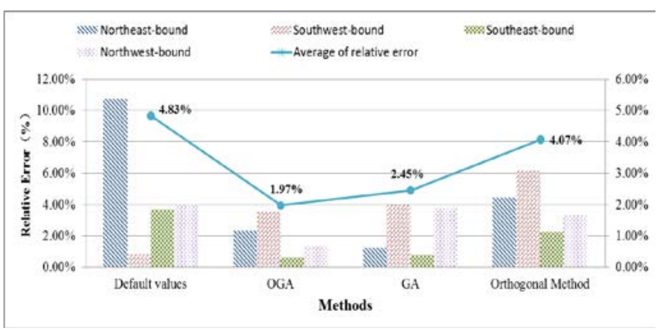


Figure 3 Comparison of TT relative error

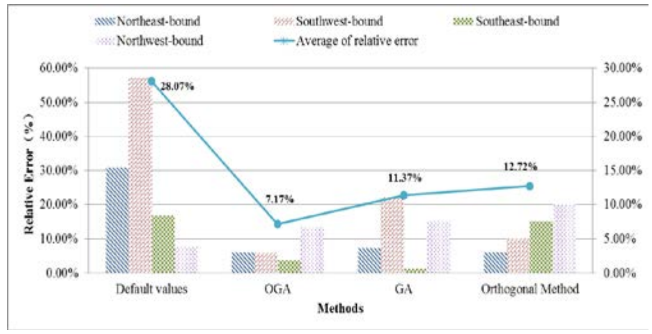


Figure 4 Comparison of MQL relative error

Fig.5 shows the comparison of convergence performance between the OGA-base calibration method and the GA-based calibration method. The former method stops when the iteration count is 10. Otherwise, the GA-based calibration method stops when iteration count is 40. To the OGA-base calibration method, it is obviously that constructing initial population consumes the most part of consuming time, because the program needs to run VISSIM 9^2 = 81 times to generate the output results. So we count the number of running VISSIM. Throughout the procedure, the proposed method runs VISSIM 238 times, and the later method runs 400 times.

Considering how the weight coefficient value impacts the calibration results. Fig.6 shows the profile of the fitness values versus weight coefficient values (i.e., weight coefficient value = 0, 0.2, 0.5, 0.8, and 1.0, respectively). The figure shows that a significant correlation exists between the fitness value and the weight coefficient values. The bigger the weight coefficient value is, the bigger the fitness value is. Considering the case of weight coefficient = 1.0, which means taking travel time into account indicator only, the fitness value is 62.70. Maximum queue length may reduce the fitness value, because field data collection exist big error.

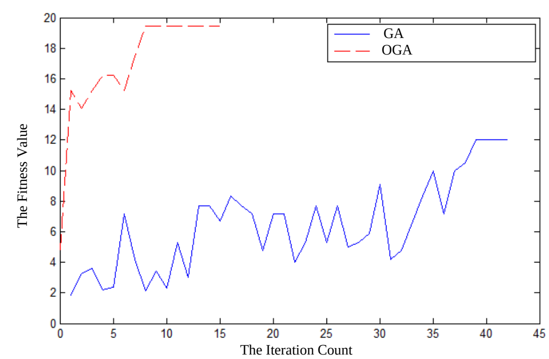


Figure 5 Comparison of convergence rate between two algorithms

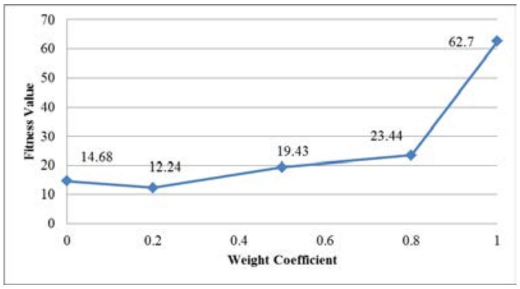


Figure 6 Comparison of the fitness value versus weight coefficient

V. CONCLUSION

This paper focuses on the automatic calibration method of traffic micro-simulation. A parameter calibration method based on orthogonal genetic algorithm is proposed. The first step, the process and the pseudo code of the OGA-based micro-simulation calibration method are given. The second step, we apply the proposed method to a signal intersection in Beijing. In the case study, the microscopic traffic flow simulation model VISSIM is selected. We compare the OGA method with the GA method and the orthogonal design method, respectively, i.e., the maximum fitness value, the relative errors of TT and MQL, the number of iteration. Experiment results show that the OGA outperforms the GA and the orthogonal design method in calibration. This paper also analyzes how the weight coefficient impacts the calibration results. A significant correlation exists between the fitness value and the coefficient values.

ACKNOWLEDGMENT

This research was supported by the National Science & Technology Support Program of China through Grant No. 2014BAG01B04.

REFERENCES

[1] D. Sun, L. Zhang, and F. Chen. “Comparative study on simulation performances of CORSIM and VISSIM for urban street network”. Simulation Modeling Practice and Theory, Vol.37, pp. 18-29, 2013

[2] H. Filmon, and P. Luis Guilherme. “Sensitivity analysis of VISSIM driver behavior parameters on safety of simulated vehicle and their interaction with operations of simulated traffic”. Transportation Research Board 92nd Annual Meeting, Washington DC, 2013

[3] F.C. Fang, and L. Elefteriadou. “Some guidenlines for selecting microsimulation models for interchange traffic operational analysis.” Journal of Transportation Engineering, Vol. 131, No. 7, pp. 535-543, 2005

[4] I. I. Otković, T. Tollazzi, and M. Šraml. “Calibration of microsimulation traffic model using neural network approach. Expert Systems with Applications”, Vol. 40, pp. 5965-5974, 2013

[5] B. Park, and J.D. Schneeberge. “Microscopic simulation modeling calibration and validation: a case study of VISSIM for a coordinated actual signal system”. Transportation Research Record: Journal of the Transportation Research Board, pp. 185-192, 2003

[6] C. Gagnon, A.W. Sadek, and A. Touchette. “Calibration potential of common analytical and microsimulation roundabout models: new England case study”. Transportation Research Record: Journal of the Transportation Research Board, pp. 77-86, 2008

[7] S. Kim, W. Kim, and L.R. Rilett. “Calibration of microsimulation models using nonparametric statistical techniques”. Transportation Research Record: Journal of the Transportation Research Board, pp. 111-119, 2005

[8] J. Asamer, H. J.V. Zuylen, and B. Heilmann. “Calibrating car-following parameters for snowy road conditions in the microscopic traffic simulator VISSIM”. IET Intelligent Transport System. Vol. 7, pp. 114-121, 2011

[9] L. Jie, H. V. Zuylen, Y. Chen, F. Viti, and I. Wilmink. “Calibration of a microscopic simulation model for emission calculation”. Transportation Research Part C, Vol. 31, pp. 172-184, 2013

[10] B. Park, and H. Qi, “Development and evaluation of a procedure for the calibration of simulation models”. Transportation Research Record: Journal of the Transportation Research Board, pp.208-217, 2005

[11] T. Ma, and B. Abdulhai. “Genetic algorithm-based optimization approach and genetic tool for calibrating traffic microscopic simulation parameters”. Transportation Research Record 1806, pp. 6-15, 2002

[12] L.O. Kim, and L.R. Rilett. “A genetic algorithm based approach to traffic micro-simulation calibration using ITS data”. Transportation Research Board 83rd Annual Meeting, Washington, D. C. 2004

[13] S. Menneni, C. Sun, and P. Vortisch. “Microsimulation calibration using speed-flow relationships”. Transportation Research Record: Journal of the Transportation Research Board, pp.1-9, 2009

[14] R. L. Cheu, X. Jin, K.C. Ng, Y.L. Ng, and D. Srinivasan. “Calibration of FRESIM for Singapore expressway using genetic algorithm”. Journal of Transportation Engineering, Vol. 124, pp. 526-535, 1998

[15] J. Ma, D. Hu, and H. Dong, and H. Zhang, “Calibration of microsimulation with heuristic optimization methods”. Transportation Research Record: Journal of the Transportation Research Board, pp. 208-217, 2007

[16] K. Aghabayk, M. Sarvi, W. Young, and L. Kautzsch. “A novel methodology for evolutionary calibration of VISSIM by multi-threading”. 36th Australasian Transport Research Forum, Australia, 2013

[17] Y. Wang, H. Liu, Z, Cai, and Y, Zhou. “An orthogonal design based constrain evolutionary optimization algorithm. Engineering Optimization”. vol.39,pp.715-736, 2007

[18] Y. Leung, and Y. Wang. “An orthogonal genetic algorithm with quantization for global numerical optimization”. IEEE Transaction On Evolutionary Computation , vol.5,pp.41-52, 2011

[19] Brackstone, M., and M. McDonald. “Car-following: a historical review”. Transportation Research Part F, pp.181-196, 1999

[20] Manijunatha, P., P. Vortisch, P, and T. V. Mathew. “Methodology for the Calibration of VISSIM in Mixed Traffic”, Transportation Research Board 92nd Annual Meeting, Washington D. C., 2013