Community Learning Analytics with Industry 4.0 and Wearable Sensor Data

István Koren and Ralf Klamma

Advanced Community Information Systems (ACIS) Group,
RWTH Aachen University, Ahornstr. 55, 52056 Aachen, Germany
{koren,klamma}@dbis.rwth-aachen.de
http://dbis.rwth-aachen.de

Abstract. Learning analytics in formal learning contexts is often restricted to collect and analyze data from students following curricula through a learning management system. In informal learning, however, a deep understanding of learners and entities interacting with each other is needed. The practice of exploring these interactions is known as community learning analytics. Mobile devices, wearables and interconnected *Industry 4.0* production machines equipped with a multitude of sensors collecting vast amounts of data are ideal candidates to capture the goals and activities of informal learning settings. What is missing is a methodological approach to collect, manage, analyze and exploit data coming from such an interconnected network of artifacts. In this paper, we present a concept and prototypical implementation of a framework that is able to gather, transform and visualize data coming from Industry 4.0 and wearable sensors and actuators. Our collaborative Web-based visual analytics platform is highly embeddable and extensible on various levels. Its open source availability fosters research on community learning analytics on a broad level.

Keywords: Community Learning Analytics, Visual Analytics, Industry 4.0, Internet of Things, Wearables

1 Introduction

Industry 4.0 refers to a paradigm shift currently taking place in industrial production towards the use of a combination of Internet and future-oriented technologies [7]. On the one hand, triggers are social, economic and political changes; on the other hand, a number of technologies like apps, 3D printers and the Internet of Things (IoT) pushes innovation in industry. Beyond the industrial context, the availability of mobile computing devices has also changed our personal lives remarkably over the last years. Smartphones, tablet computers and smart watches have become commodities and are used for various use cases. Both Industry 4.0 appliances and personal smart devices are equipped with a multitude of sensors that produce a lot of data. Wearable computers build the vanguard, with sensors that are tightly integrated with body functions, like heart rate sensors and eye trackers. In this context, the term Internet of Things represents the idea that everyday devices become interconnected to form a huge network of artifacts that is closely embedded into social interactions. However, the speed of innovation is currently hampering the adoption of a compatible standard; the challenges lie in the sheer number of devices, protocols, standards and platforms.

For technology enhanced learning researchers, the broad availability of IoT technologies in Industry 4.0 and wearable contexts opens new doors to explore the possibilities and limitations of applying wearables for various workplace learning settings. Research questions include how to leverage body and device sensors and, more generally, contextual information to provide and sustain adequate services to learners. Traditional formal learning analytics often targets the interactions of learners with learning management systems while neglecting the context and environment of learners. In informal learning contexts, goals and activities are not fixed in a curriculum; they may be more short-term [5]. While white-collar knowledge worker communities such as the insurance claims processors described by Wenger [11] leave analyzable traces in the software they are using, the digital footprints of industrial blue-collar machine workers are typically more fragmented and numerous across machines and wearable sensors. We realize that methods known from formal learning analytics are not applicable for the vast amount of heterogeneous data sources available from sensors. With new types of sensors and subsequently new kinds of data available on a regular basis, we need a cross-cutting methodological and sustainable approach for targeting all kinds of possible scenarios. In this paper we therefore present the Social Web-based Environment for Visual Analytics (SWEVA), a conceptual approach and prototypical implementation of services able to retrieve and visualize data from heterogeneous data sources such as machine data or wearable sensors. It is extensible on various levels to handle a wide variety of current and future protocols and standards in the IoT realm. Within the framework, advanced social network analysis tools can be accessed, such as overlapping community detection and expert identification to significantly drive forward community evolution. Due to its Web-based nature, learners are able to open the application in any browser to take part in their communities' analytical undertakings. Ultimately, our platform for analyzing learning services for Industry 4.0 and wearable sensors may help to significantly increase the relevance and applicability of learning services in this domain.

The paper is organized as follows. First, we give the motivation for our research in Section 2 and highlight related work in Section 3. We then present the concept in Section 4. Section 5 describes the prototypical implementation that is evaluated in Section 6. The paper is concluded in Section 7 with an outlook on future work.

2 Motivation

Learning with Industry 4.0 appliances and wearables is different from traditional classroom based learning in various ways. Mainly, it is not tied to a particular location and time. It may leverage the actual context of the user, spanning from detecting the actual physical location and even tool the user is employing, up to measuring body parameters like the current heart rate. Industry 4.0 appliances and wearables offer an enormous amount of data and are therefore usable by a huge variety of learning services. Examples of such data are inventory trackers and alerts on incorrect operation. In comparison, body-worn wearable sensors may capture the heart rate, arm movements or track the eye gaze. However, the more data is available, the more complex it is to analyze and reason upon it. What is needed is a uniform approach for (real-time) learning analytics. One of the challenges in reasoning and researching on this data lies in the high degree of context sensitivity and interdependency of data coming from machine and human data sources. For example, a higher stress level identified through a significantly increased heart rate may be the cause or effect of machine malfunction. On the technical level, currently different implementations of sensor networks often struggle with a myriad of standards for accessing the data and related intercompatibility problems. For instance, standards and protocols for IoT include MQTT, XMPP, CoAP, Bluetooth, and Zigbee, amongst many others. However, we are observing a consolidation towards open Web protocols to make the data available on the human-facing side. Specifically, this means that while the actual machine-to-machine communication happens over proprietary protocols, most commercial off-the-shelf solutions come with a gateway that translates device-specific communication channels to the open HTTP Web standard, so that it can be accessed by apps running on smartphones. That is the main point of contact of our framework for getting device data into our analytics pipeline to perform visual analytics tasks.

Visual analytics (VA) is the "science of analytical reasoning facilitated by interactive human-machine interfaces" [4]. It is a multidisciplinary approach covering data science, data management, data analysis, human computer interaction and decision support amongst many others. The VA process is displayed in Figure 1. It starts with collecting and optionally transforming data. The goal is to gain knowledge through building models and visualize them. Thereby, the visual part remains highly adaptable based on user interaction. Knowledge can be used to adapt and improve the monitored artifacts, or to calibrate the selection of data sources.

In informal learning contexts, there is a lack of institutional rules which induces the negotation of roles in communities based on reputation and expertise [5]. Community learning analytics is therefore concerned with identifying expertise within a community and in comparison with other communities. A learning system able to discover the experts within a community is empowered to transfer knowledge from experienced users to new staff. For instance, wearable sensors may capture the expert fulfilling a certain task by operating a machine. Later, the recordings may be replayed to less experienced users through augmented reality devices. The access to expert identification [12] and expert recommender algorithms [2], along with the visualizations of their outputs, is therefore a crucial requirement for our system. (Overlapping) community detection algorithms in turn are key to expert identification systems, as they are able

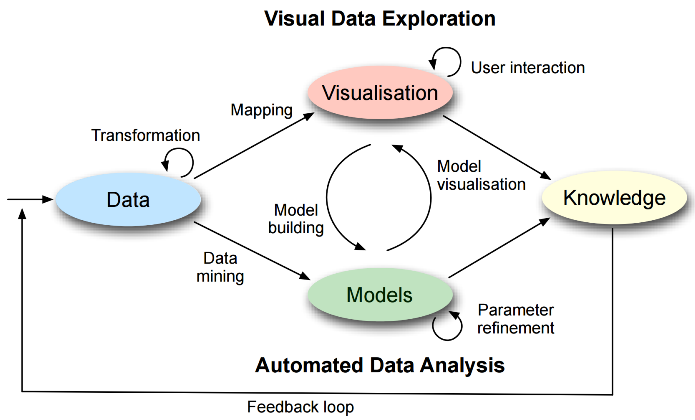

Fig. 1. Visual Analytics Process [4]

to distinguish between the core and the periphery of a community. Moreover, the propositions of visual analytics are twofold. On the one hand, the effectiveness of the measured knowledge transfer may be visualized on a dashboard. On the other hand, visual cues gained through analytics can themselves be embedded right into the performance augmentation process. For instance, augmented reality devices may integrate security alerts coming from self-managed machines in the field of vision.

Our methodology presented in the following reaches from the gathering and transformation of data coming from innovative wearable-based learning services up to the support of its usage via (near) real-time visual analytics.

3 Visual Analytics Platforms for Internet of Things Data

In this section, we reference related research and commercial work for visual analytics of heterogeneous IoT data. In particular, we look at Web-based dashboards to make device-specific data available for evaluation on the Web.

IBM Watson IoT Platform is a commercial visual analytics platform for the Internet of Things [3]. The Web application offers IoT analytics features for device management and analytics applications. Devices can be configured to send data into the cloud using the MQTT protocol. Applications can then interact with the data. Finally, collected data can be visualized in a configurable dashboard that provides location, live property values as well as alerts caused by user-defined rules.

Bosch provides their own IoT solution called the Bosch IoT Suite [1]. Using the provided development toolbox, users can create their own IoT applications. The platform offers an IoT Hub component that transports IoT data from sensors to the applications developed by the customer.

Pheme is a cloud-based service that repurposes Web analytics services for IoT data collection and visualization [8]. Web analytics usually refers to the collection and evaluation of data that users produce when visiting websites. In their work, Mikusz et al. mapped typical Web analytics properties to IoT events. Pheme consists of four different modules: import, preprocessing, visualization and reporting.

We presented three representatives of visual analytics systems that process data coming from heterogeneous data sources. In the area of learning analytics for informal learning, we mainly find approaches displaying data in a preconfigured dashboard. A representative of this research is Social Semantic Server Dashboard by Ruiz-Calleja et al. [10]. It is able to collect and visualize data collected from Social Semantic Server, a semantically-enriched artifact-actor network. What remains unclear, is the applicability of the dashboard to accommodate dynamic real-time data.

What became evident in our research survey is the lack of tools specifically designed for learning analytics for heterogeneous data sources. In particular, we miss approaches integrating social network analysis methods like (overlapping) community detection and expert identification. The existing solutions further do not support the dynamic export and recomposition of the visualization widgets into third-party websites. In the following, we present our conceptual approach to fill these gaps.

4 Data Processing and Analytics Pipeline

The main building block and reasoning behind our approach is to leverage open Web technologies over the whole chain from accessing device data to analyzing it. Our system collects data from heterogeneous sources through Web technologies and makes it available via a browser-based platform that visualizes the data in near real-time. The complete pipeline and its layers are shown in Figure 2 from top to bottom in chronological order. The platform is extensible in terms of more data sources and visualization options. In the following, we explain the three main additions of our framework in detail, namely the core framework, the model editor and the visualization frontend.

As shown in Figure 2, the pipeline starts at the data source level. Data sources can be anything from real Industry 4.0 machines, body-worn wearable sensors or input captured on smartphones. Besides, data can also be retrieved from any website, such as open data provided by governmental and non-governmental organizations. The actual data format like JSON or XML is not yet important in this step.

The next level is the data aggregation tier. Here, data that is typically not available over the Internet is made available on repositories. For instance, sensors that are using heterogeneous exchange protocols may upload their capturings to a common IoT database. This step also comprises 3rd party services, such as sentiment detection of discussion forums or map providers.

After the first steps that make the data available on the Web, the collaborative model editor comes into play. It is a tool for creating and editing visualization pipeline models that define the processing steps from raw data to highly

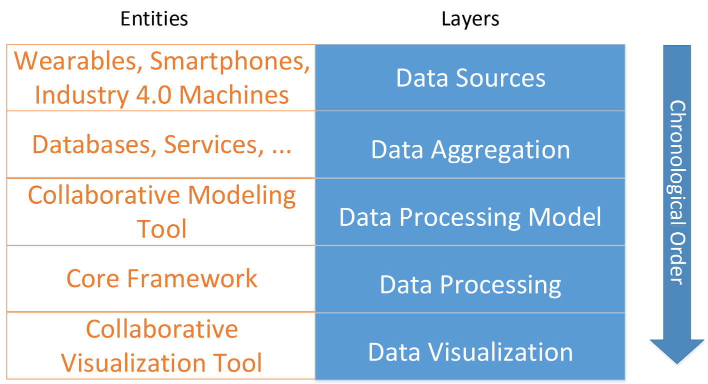

Fig. 2. Data Processing Pipeline

interactive visualizations. These models are graphs consisting of nodes and edges connecting the nodes. Per definition, these need to be directed acyclic graphs (DAGs). DAGs consist of a finite number of nodes and edges with each edge directed from one node to another, without any cyclic loops. There are two types of nodes available in the collaborative model editor; first, data processing nodes and second, input nodes. Data processing nodes can either retrieve raw or processed data from a data source, e.g. a REST-based Web interface or a Websocket-based push server, or they perform a calculation, e.g. transforming data. Input nodes represent parameters that influence the data retrieval or visualization options. They represent the screws to be adjusted during the visualization. An example is filtering which data is shown in a visualization.

The data pipeline remains open for continuous refinement during the visual analytics process. Created models can be grouped into compositions and exported for later use. That way, recurring, complex data transformations only need to be modeled once; later, they can be imported into other visualization pipelines.

Once the model is ready, the core framework comes into play. The core framework is responsible for running the pipelines defined in the model. When a model is executed for the first time, the core framework collects the default values of the user input nodes and then runs the code within the data processing nodes. It collects and transforms data using Web services or local computations. Multiple modules within a model can be executed concurrently. At the end, it calculates the final output of the model execution and makes it available to the visualization. The visualization tool is responsible for displaying the results of the model execution. It provides user interface elements for editing the user input variables. Upon changes to the user input, it calls the core framework to recalculate the results, which in turn triggers the recalculation of the visualization. In the next section, we explain the prototypical implementation of the Web-based system for visual analytics.

5 Social Web-Based Environment for Visual Analytics

For the prototype implementation, we use a component-based software architecture. The frontend is developed using the state-of-the-art Web components group of W3C standards. Similar to the efforts of earlier widget-based frontends, Web components allow the definition and encapsulation of user interface widgets into reusable software packages. They can later be embedded into arbitrary websites, keeping their functionality. For executing long-running data retrieval and transformation operations, we set up a microservice-based backend. Like Web components on the frontend, microservices encapsulate well-described functionalities into their own software packages for later reuse in a different context. Figure 3 shows a screenshot of the whole Web application. The instance portrayed in the figure displays the processing model for retrieving development data from an open source repository on GitHub. Specifically, the model first retrieves raw data in JSON format from a public Web service. The dataset contains dates as index and the number of lines added or removed as values. This dataset is then separated into additions and deletions via two processing nodes. Finally, the data is fed into a line chart that can be seen on the right of the figure. In the following, we explain the details of the implementation based on the conceptual parts described in the section above.

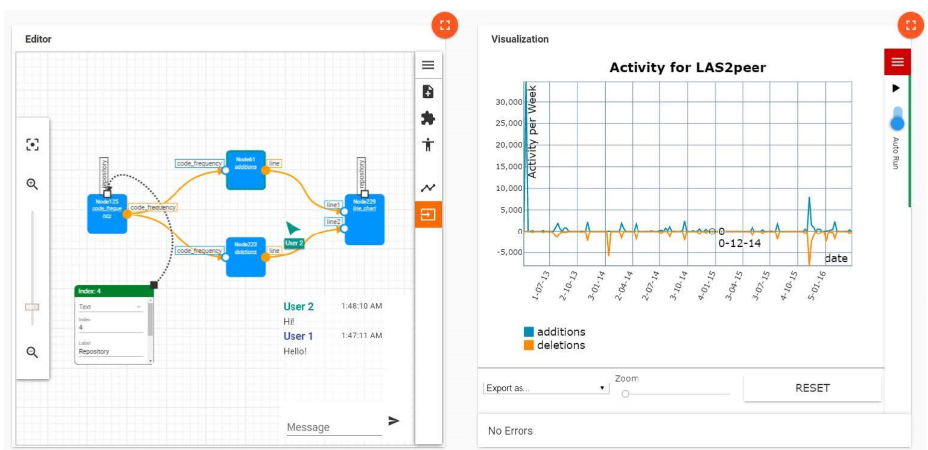

Fig. 3. Screenshot of the Web-Based Visual Analytics Tool

The collaborative model editor is implemented as a standalone Web component, thus it is embeddable into arbitrary websites. The interface consists of a model viewer and editor, which we developed using the open source jsPlumb Toolkit[1]. The jsPlumb Toolkit enables developing graph-based modeling application for Web browsers. On the left side, we added zoom controls to be able

[1] https://jsplumbtoolkit.com/ last accessed in April, 2017

to look into details of the model. On the right side, a toolbar gives access to various model-related functionalities, like adding input nodes or data processing nodes. To simplify reuse, we added a library of pre-defined data processing nodes which can be browsed through in the interface. The visualization types, e.g. line chart, stacked area chart or bar chart, can be selected in the toolbar as well. Finally, the update model button generates the data visualization pipeline as JSON document and hands it over to the core framework.

The core framework is responsible for running the data retrieval and transformation pipeline. It is developed in JavaScript and can thus run in browser or server environments, the latter using a NodeJS instance. We also developed an execution service to be able to run it within our Java-based peer-to-peer microservice framework called las2peer [6]. After the core framework has generated an output JSON file, it is handed over to the visualization tool.

The visualization tool consists of a viewer where various charts can be loaded and displayed. It inherently supports zooming and panning operations; besides, the diagrams it loads may offer further functionalities like expanding or collapsing a tree structure or limit the displayed key space to a certain time period. Similar to the model editor, the right side features a toolbar. It contains the controls defined as user inputs. Currently, we support text, number, numerical slider, toggle, dropdown and fixed value inputs. The inputs are automatically validated according to their type. For instance, user inputs to numerical fields are checked whether they represent numbers; if not, they are not processed. In case of such a validation error or other malfunctions during running the pipeline, an error is displayed in a logging pane at the bottom of the screen.

As stated above, both frontend parts are developed using Web components. One of the main advantages are their reusability. After importing and declaring them in HTML, they can be used as normal elements on the page. We use the Polymer[2] library from Google, as it adds compatibility to various browsers, some syntactical sugar, and most importantly, a consistent set of pre-designed user elements adhering to the Material Design guidelines[3]. This enabled us focussing on functionalities, rather than browser quirks and accessibility issues.

6 Preliminary Evaluation

To validate our results, we performed preliminary technical and usability evaluations of our system. For reproducibility reasons, we used an IoT dataset from hurricane Katrina [9], one of the most severe natural disasters in the history of the United States. The dataset contains multiple thousand measurements of several weather stations. In our scenario, the data was replayed through an XMPP server, and our visual analytics frontend was connected to the XMPP server. For up to 30 nodes, the near real-time graph widget rendered the graph at a speed of around 55 frames per second. When adding more than 30 nodes, the visualization began to slow down. For around 70 clients, the frame rate dropped

[2] https://www.polymer-project.org/ last accessed in April, 2017
[3] https://material.io/ last accessed in April, 2017

to around 30 frames per second. When monitoring a network with 110 clients, we still measured 20 frames per second.

We additionally invited 12 volunteers out of our pool of bachelor and master students of computer science and performed a usability study. In total, we held six evaluation sessions with two participants in each. The participants were asked to collaboratively generate near real-time visualizations using the IoT dataset described above. We provided two laptops running on Windows 10, with recent Chrome browsers. A third laptop hosted the frontend and backend services as well as the evaluation network simulation. After the modeling of the data pipeline, the participants were asked to identify certain nodes in the analytics. For that, they had to interact with the visualization. Finally, the users had to fill in a survey. Although most participants knew about the Internet of Things paradigm, only few were familiar with the details of IoT protocols and visual analytics. All users agreed that for the given analytics tasks, extracting the information via the graphs was efficient and easily comprehensible. Furthermore, the availability of near real-time visualizations was helpful in understanding the inner working of the IoT network. Minor usability issues detected in our tests like finding the right buttons could be solved by changing the icon and offering tooltips. Overall, the preliminary evaluation showed the usefulness particularly in scenarios where a large set of data is available.

7 Conclusion and Future Work

The Internet of Things, Industry 4.0 and in particular wearable computing are currently introducing a new era of ubiquitous computing. For researchers in technology-enhanced learning, this opens the door to a whole new way of analyzing learners' behaviors especially in informal learning settings. By reading and interpreting sensor data in near real-time, learning services can be adapted and continuously improved more precisely. Yet what is missing is a cross-device infrastructure for setting up these kind of community learning analytics services. In this paper, we presented a highly flexible approach to visualize live data coming from IoT sensors in Industry 4.0 contexts. Our simple Web-based visual analytics tool is able to capture and transform data coming from a wide variety of sources and formats. It can be arbitrarily extended both in the execution phase and in the visualization parts. We performed an initial evaluation from the technical and usability perspectives that lead to promising results.

On the practical side, we are working on a library of prepackaged modules to cover a wide variety of data sources. Already, we provide modules for retrieving JSON, XML, MQTT and XMPP protocol data, covering a majority of IoT gateways and learning management system APIs. Current challenges include taking into consideration aspects like data quality, and showing uncertainty in the visualizations. As we developed the concept and implementation, we neglected privacy aspects to a large part, thus ethical and moral issues need to be discussed alongside evaluations. Possible future tasks include machine learning techniques to further automate analytics tasks. The tools are available open source on our

GitHub repository[4]. We would like to open the academic discussion to research new use cases in the area of informal learning.

Acknowledgements. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme project "Learning Layers" (grant no. 318209) and the European Union's Horizon 2020 Programme through the project "WEKIT" (grant no. 687669).

References

1. Bosch: Bosch's IoT platform (2017), https://www.bosch-si.com/de/iot-plattform/bosch-iot-suite/homepage-bosch-iot-suite.html
2. Drachsler, H., Hummel, H.G.K., Koper, R.: Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model. International Journal of Learning Technology 3(4), 404–423 (2008)
3. IBM: Watson IoT Platform (2017), https://www.ibm.com/internet-of-things/platform/watson-iot-platform/
4. Keim, D.A., Andrienko, G., Fekete, J.D., Görg, C., Kohlhammer, J., Melançon, G.: Visual Analytics: Definition, Process, and Challenges. In: Kerren, A., Stasko, J., Fekete, J.D., North, C. (eds.) Information Visualization, LNCS, vol. 4950, pp. 154–175. Springer Berlin / Heidelberg (2008)
5. Klamma, R.: Community Learning Analytics – Challenges and Opportunities. In: Wang, J.F., Lau, R.W.H. (eds.) Advances in Web-Based Learning: ICWL 2013. LNCS, vol. 8167, pp. 284–293. Springer, Berlin (2013)
6. Klamma, R., Renzel, D., de Lange, P., Janßen, H.: las2peer - A Primer (2016)
7. Lasi, H., Fettke, P., Kemper, H.G., Feld, T., Hoffmann, M.: Industry 4.0. Business & Information Systems Engineering 6(4), 239–242 (2014)
8. Mikusz, M., Clinch, S., Jones, R., Harding, M., Winstanley, C., Davies, N.: Repurposing Web Analytics to Support the IoT. Computer 48(9), 42–49 (2015)
9. Patni, H., Henson, C., Sheth, A.: Linked Sensor Data. In: 2010 International Symposium on Collaborative Technologies and Systems. pp. 362–370
10. Ruiz-Calleja, A., Dennerlein, S.M., Ley, T., Lex, E.: Visualizing workplace learning data with the SSS Dashboard. In: CrossLAK 2016 Learning Analytics Across Physical and Digital Spaces 2016, CEUR Workshop Proceedings, vol. 1601. CEUR (2016)
11. Wenger, E.: Communities of Practice: Learning, Meaning, and Identity. Learning in doing, Cambridge University Press, Cambridge, UK (1998)
12. Worsley, M., Blikstein, P.: What's an Expert? Using learning analytics to identify emergent markers of expertise through automated speech, sentiment and sketch analysis. In: Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero, C., Stamper, J. (eds.) 4th International Conference on Educational Data Mining. pp. 235–240 (2011)

[4] https://github.com/rwth-acis/SWeVA-Editor-Page/