ACCSE 2016

The First International Conference on Advances in Computation, Communications and Services

ISBN: 978-1-61208-481-7

May 22 - 26, 2016

Valencia, Spain

ACCSE 2016 Editors

Sofiane Hamrioui, University of Haute Alsace, France
Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

ACCSE 2016

Foreword

The First International Conference on Advances in Computation, Communications and Services (ACCSE 2016), held between May 22-26, 2016, in Valencia, Spain, is targeting the progress made in computation, communication and services on various areas in terms of theory, practices, novelty, and impact. Current achievements, potential drawbacks, and possible solutions are aspects intended to bring together academia and industry players

The rapid increase in computation power and the affordable memory/storage led to advances in almost all the technology and services domains. The outcome made it possible advances in other emerging areas, like Internet of Things, Cloud Computing, Data Analytics, Smart Cities, Mobility and Cyber-Systems, to enumerate just a few of them.

We take here the opportunity to warmly thank all the members of the ACCSE 2016 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to ACCSE 2016. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the ACCSE 2016 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that ACCSE 2016 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of computation, communication, and services.

We are convinced that the participants found the event useful and communications very open. We hope that Valencia provided a pleasant environment during the conference and everyone saved some time to enjoy the charm of the city.

ACCSE 2016 Chairs:

ACCSE Advisory Committee

Carlos Becker Westphall, Federal University of Santa Catarina - Florianópolis, Brazil
Sofiane Hamrioui, University of Haute Alsace, France
Darshika G. Perera, University of Colorado at Colorado Springs (UCCS), USA
Sergio Ilarri, University of Zaragoza, Spain
Shigeaki Tanimoto, Chiba Institute of Technology, Japan
Yanfang Ye, West Virginia University Morgantown, USA
Thierry Rakotobe-Joel, Ramapo College of New Jersey | Anisfield School of Business, USA

ACCSE 2016

COMMITTEE

ACCSE Advisory Committee

Carlos Becker Westphall, Federal University of Santa Catarina - Florianópolis, Brazil
Sofiane Hamrioui, University of Haute Alsace, France
Darshika G. Perera, University of Colorado at Colorado Springs (UCCS), USA
Sergio Ilarri, University of Zaragoza, Spain
Shigeaki Tanimoto, Chiba Institute of Technology, Japan
Yanfang Ye, West Virginia University Morgantown, USA
Thierry Rakotobe-Joel, Ramapo College of New Jersey | Anisfield School of Business, USA

ACCSE 2016 Technical Program Committee

Ali Akdagli, Mersin University, Turkey
Markus Aleksy, ABB AG, Germany
Cristina Alcaraz, University of Malaga, Spain
Carlos Becker Westphall, Federal University of Santa Catarina - Florianópolis, Brazil
Siegfried Benkner, University of Vienna, Austria
Jacir L. Bordim, University of Brasilia, Brazil
Marc Bosch, Qualcomm, Inc., San Diego, USA
Anastasios A. Economides, University of Macedonia, Greece
Magdy A. El-Moursy, Electronics Research Institute, Cairo, Egypt
Gabriel Falcao, IT / University of Coimbra, Portugal
Hong Fu, Chu Hai College of Higher Education, Hong Kong
Veronica Gil-Costa, Universidad Nacional de San Luis, Argentina
Ki-Il Kim, Gyeongsang National University, South Korea
Sofiane Hamrioui, University of Haute Alsace, France
Hiroaki Higaki, Tokyo Denki University, Japan
Sergio Ilarri, University of Zaragoza, Spain
Ilias Iliadis, IBM Research - Zurich, Switzerland
Atsushi Kanai, Hosei University, Japan
Keiichi Kaneko, Institute of Engineering - Tokyo University of Agriculture and Technology, Japan
Kyungtae Kang, Hanyang University ERICA, South Korea
Christos Kartsaklis, Oak Ridge National Laboratory, USA
Kenji Kise, Tokyo Institute of Technology, Japan
Zbigniew Kotulski, Warsaw University of Technology, Poland
Nurul Mu'azzah Abdul Latiff, Universiti Teknologi Malaysia, Malaysia
Min Luo, Huawei Technologies, China
Michele Magno, ETH Zurich, Switzerland
Alan Marchiori, Bucknell University, USA
Susumu Matsumae, Saga University, Japan
Hidemoto Nakada, National Institute of Advanced Industrial Science and Technology, Japan
Rafael Pasquini, Federal University of Uberlândia, Brazil
Darshika G. Perera, University of Colorado at Colorado Springs (UCCS), USA

Copyright Information

Table of Contents

Collaboration Web - Social Computing Technology for Emergency Data Interoperability 1
Aurel Machalek and Dominic Dunlop

Towards Drone-Assisted Large-Scale Disaster Response and Recovery 5
Michael Spranger, Florian Heinke, Sven Becker, and Dirk Labudde

In-Vehicle Interface Assessment Framwork for Emerging Technologies: A Case Study for Evaluating New Driver-vehicle Interaction Design 11
Myoungouk Park, Bawul Kim, and Joonwoo Son

Real-Time Detection and Classification of Driver Distraction Using Lateral Control Performance 13
Joonwoo Son and Myoungouk Park

Evolution of Automated Regression Testing of Software Systems Through the Graphical User Interface 16
Pekka Aho, Emil Alegroth, Rafael Oliveira, and Tanja Vos

Collaboration Web - Social Computing Technology for Emergency Data Interoperability

Aurel Machalek, Dominic Dunlop
University of Luxembourg
Interdisciplinary Centre for Security, Reliability and Trust
Luxembourg, Luxembourg
email: aurel.machalek@uni.lu, dominic.dunlop@uni.lu

Abstract— **This paper discusses the use of social media for data interoperability and sharing between agencies during emergency situations, natural disasters, and disruptions of critical infrastructure. The work presented is ongoing in the context of the Emergency Responders Data Interoperability Network (REDIRNET) research project. The project's purpose is to provide a decentralized framework for interoperability between emergency agencies' systems, based on a public meta-data gateway controlled by the agencies through a socio-professional web interface. A major problem during crisis situations is sharing data from various sources between the agencies involved. These data are often heterogeneous and distributed between many organisations with different access rights, and may have little or no security level protection. This paper focuses on the shared platform, which collects information from a variety of sensor nodes and presents it in a user-friendly manner. This is a new approach to using social networks actively in the field of Public Protection and Disaster Relief by addressing the technological challenge through an open-sourced metadata gateway combined with social computing technology.**

Keywords-Collaboration web; Data interoperability; Social computing technologies.

I. CONTEXT AND MOTIVATION

In recent years, First Responder organizations across Europe have considerably improved their communications and IT systems through the deployment of new technologies, including such innovations as unmanned surveillance and sensor systems that assist in preventative actions and enhance responses to major crisis events. Nevertheless, a number of recent major incidents have highlighted the challenges first responders face, most notably concerning barriers to interoperability [1], [2]. These challenges must be faced despite the current economic and financial situation, which places agencies under considerable budgetary pressure, leaving them unable to invest significant sums in enhancing their interoperability capabilities – particularly when such cooperation between agencies is thankfully not required frequently. This leads to the conclusion that if we are to enhance agency interoperability, this must be achieved through cost-effective solutions. Interoperability is especially important during major incidents, when many agencies are involved. In addition to the core responder agencies, major crisis events invariably involve public utilities, technical

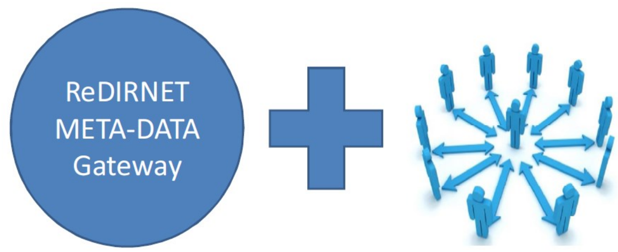

Figure 1. Metadata gateway combined with socio-professional networking.

rescue, the coastguard, search & rescue, highway management, hazmat and other disaster relief teams. As a consequence, several command posts and crisis management centres are in operation and need to inter-operate. The need for interoperability between different agencies has often led to the development of one-to-one interconnection solutions, which suffer from maintenance issues as the connected systems evolve. Moreover, such ad hoc solutions often fail to consider emerging concerns related to security and privacy.

Two main technologies are combined in our project: a *common metadata gateway* and a *socio-professional networking system* [3], [4], [5], [6], [7], allowing each agency to set the visibility and controllability of its data for each partner agency and each data field, as shown in Figure 1. The project introduces a system that provides seamless interoperability for participating agencies at virtually zero cost, while still offering great flexibility as regards what data can be available to partner agencies via the socio-professional web. This level of interoperability offers emergency service agencies a more effective response to major crisis incidents that may ultimately lead to enhanced safety and security for the public across Europe.

In the remainder of this paper, we first discuss related work in Section II, before moving in Section III to a presentation of the REDIRNET components used to implement the collaboration web described in Section IV. Our conclusion is presented in Section V.

II. RELATED WORK

The idea of the Collaboration Web took shape in 2012, with the launch of the Free Secure Interoperable Communication (FREESIC) project [8]. Its goal was to

 ISBN: 978-1-61208-481-7

"allow highly secure and cost effective interoperability between communication infrastructures over the entire Europe". The approach chosen was to leverage existing interoperability solutions such as gateways, simplifying FREESICs adoption by agencies, and, in return, opening broader possibilities for them. The service operated free of charge and offered open source gateway code, documentation and operational guidelines for others to use. Provision was made to continue the free-of-charge operation after the project's end. The architecture took into account ongoing standardization research (e.g., the **Network Centric Operations Industry Consortium** (NCOIC) Interoperability Framework [9]) to reduce integration time and costs. The integration process was simple: the system integrator took the gateway equipment and modified it as needed. While the gateway remained the property of the integrator, the integrator did not have to worry about disclosing any know-how or information. Communication between gateways was end-to-end encrypted and each gateway was under the full control of the end user, so as to avoid security concerns2. The project was successful in showing that the initial concept of the Collaboration Web was workable.

III. REDIRNET COMPONENTS

The main focus of the current project [10] is to create possibilities for interagency communication and the sharing of first responders' data. Our system allows an agency to access a partner's shared sensor data. Based on the sensor and its capability, two kinds of connection can be set up: a request/response connection is used for batched data, while a stream connection is established for a continuous data flow. It is assumed that the partner agency consuming the data is capable processing the incoming data. As an example, in Figure 2, Agency B is offering data streams from Sensor 4 and Camera 1 to Agency A. This arrangement has been previously agreed between the two agencies and configured using the REDIRNET socio-professional network. The network's seamless interoperability means that Agency B's Sensor 4 and Camera 1 can be displayed in agency A's control center beside its own Sensor 6 and Camera 3. The same is true for data displayed on Agency A field officers' handheld devices.

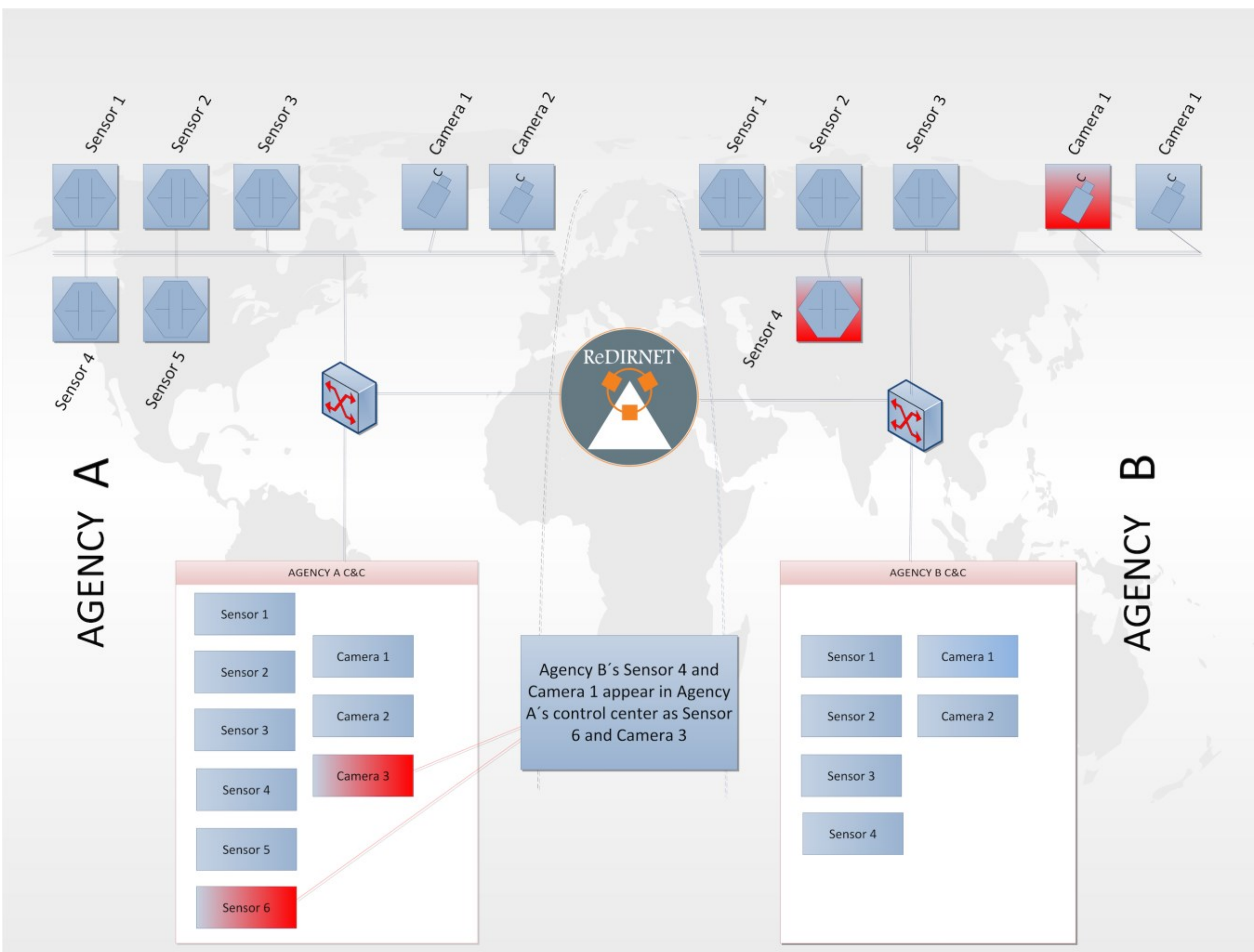

Figure 2. REDIRNET components.

 ISBN: 978-1-61208-481-7

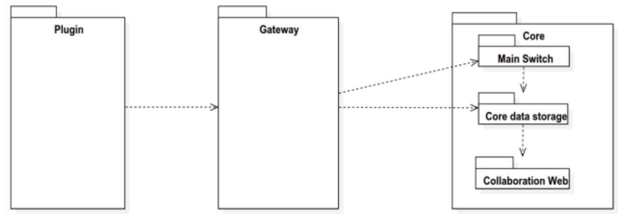

Figure 3. Class diagram showing REDIRNET components.

In order to provide agencies with the ability to share their data resources, some preliminary steps are required. The data-producing resource must be connected to some sort of translator, which can encode the data stream into a common transport protocol, and provide the correct command interface. This translator is then connected to a transport network, which can verify permissions for each user and route the stream to its destination, where it is translated into a protocol native to the data consumer and subsequently displayed.

This workflow, shown in Figure 3, can be achieved by a system consisting of five components:

- ***Main Switch***, redirecting the communication, checking permissions and providing a logging facility,
- ***Core Data Storage*****,** supplemented by an ontological search engine, providing database services for all data requiring storage,
- ***Collaboration Web***, the user interface for the system, allowing resources to be registered and their permissions to be managed,
- ***Gateways***, a client at each agency client, taking the role of mediator between the Main Switch and the plug-ins,
- Plug-ins, drivers providing interfaces for the endpoint resources.

IV. Collaboration Web

The socio-professional networking component provides a decentralized opportunity for an interoperability network to be built and configured by its users. It also allows the interoperability network to be run without major operational costs, since the collaboration rules are set by the agencies themselves by following basic guidelines. Collaboration rules are set according mutual agreements between the agencies involved and should cover issues that include the visibility and controllability of data fields, data streams and switches. The algorithms and keys used for end-to-end encryption of the data are also set up by the agencies.

The Collaboration Web handles the use cases related to getting in touch with other agencies and the high-level configuration of the inter-connection based on abstract roles. Typical use cases are the creation of an agency profile, registration of a data resource, and configuration of permissions. The agency's authorized users can perform all

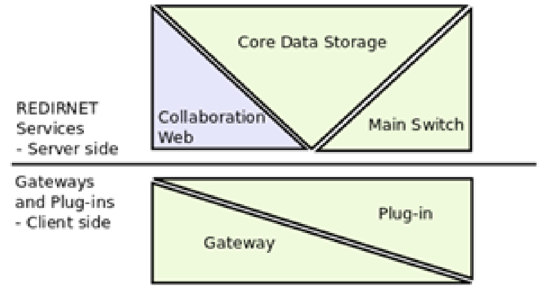

Figure 4. Collaboration Web system in relation to other components.

these actions through a web browser. The web interface will initially provide all of the essential collaboration capabilities mentioned above, but, should the need arise, it can be extended to provide additional functionalities of the type found today in commercial applications.

In line with modern approaches to the development of web applications, we are using the Model-View-Controller programming paradigm for enhanced robustness and modularity, as shown in Figure 4. The services of the REDIRNET Core, particularly its Core Data Storage component, will implement the Model and Controller functions of the paradigm, while the Collaboration Web itself will provide the View role.

Figure 5 shows the operations allowed by the collaboration web, an example screen for which appears as figure 6. These operations will allow each agency's authorized users to:

- register users and maintain profiles;
- enter name, short-name, description, and contact information;
- establish partnerships between agencies;
- search for a partner agency, propose/approve a partnership, display the partners of an agency and whether a partnership was mutually approved;
- search for the data resources of other registered agencies;

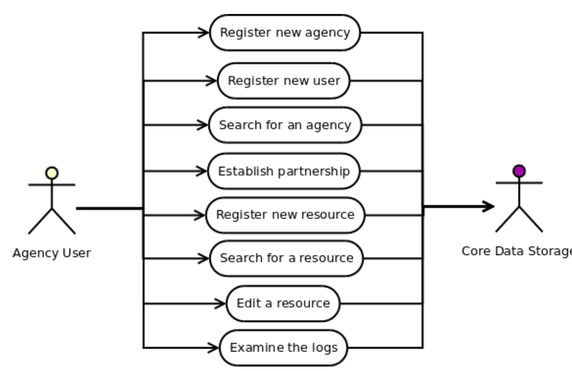

Figure 5. Collaboration Web system operations.

 ISBN: 978-1-61208-481-7

Figure 6. Collaboration Web ontology search (detail).

- set up interoperability for resources with the help of the ontology subsystem, which allows the user to use the terminology native to their region or domain; and
- examine log events and set the correct permissions on each resource.

The Collaboration Web should also be able to initiate a crisis event, which makes it possible to override some of the permissions. (This possibility of overriding existing resource permissions will be of course based on agreement between the respective agencies.) A major issue in such collaborative systems is data privacy, especially where international cooperation is involved. To solve such concerns, the collaboration system will be available only to agencies in the field of Public Public Protection and Disaster Relief, and subject to bi- or multilateral agreements based on trust relationships estabished before each agency joins the system.

V. CONCLUSION

The core features of the REDIRNET system have already been tested and the system is being prepared for integration by the first test users. These core features are related to data transfer from the producing plug-in and gateway, via the main switch, to the consuming gateway and plug-in. The system also supports a limited set of social networking features, semantic search and agent-assisted resource polling. The system remains under development, and we expect to implement other features defined in the system requirements in the coming months. The whole system will be evaluated in the final demonstration, which will include both emergency agencies and end users. The usability of the system must satisfy both technical (functional) and non–technical requirements, such as process/culture, financial & commercial, legal & regulatory, and data privacy, which will be developed in cooperation with end users at the beginning of the research project.

ACKNOWLEDGMENT

This work is supported by European Union projects FP7-607768 REDIRNET: Emergency Responder Data Interoperability Network, VEGA No. 2/0054/12 and CLAN APVV-0809-11

REFERENCES

[1] G. Baldini, "Report of the workshop on 'Interoperable communications for Safety and Security'," European Commission Joint Research Centre Institute for the Protection and Security of the Citizen, 2010. [Online]. Available from: http://rp7.ffg.at/upload/medialibrary/lbna24540enc.pdf [retrieved: April, 2016]

[2] R. Mitchell, C. Eckel, and J. Mathis, "Implementation Profile for Interoperable Bridging Systems Interfaces." NIST/OLES, 2009. [Online]. Available from: http://www.firstresponder.gov/Lists/Safecom/BridgingSystemsInterfaceCoreProfile11_FINAL.pdf [retrieved: April, 2016]

[3] N. Huijboom et al., "Public Services 2.0: Social computing and its implications for future public services," Institute for Prospective Technological Studies, 2009. [Online]. Available from: http://ftp.jrc.es/EURdoc/JRC54203.pdf [retrieved: April, 2016]

[4] C. Pascu, D. Osimo, M. Ulbrich, G. Turlea and J. C. Burgelman, "The potential disruptive impact of Internet 2-based technologies," First Monday, vol. 12, no. 3, 2007. [Online]. Available from: http://firstmonday.org/article/view/1630/1545 [retrieved: April, 2016]

[5] M. Slot and V. Frissen, "Users in the 'golden' age of the information society," Observatorio Journal, vol.1, no. 3, pp. 201–224, 2007.

[6] M. Madden and S. Fox, "Riding the waves of 'Web 2.0'," Pew Research Center, 2006. [Online]. Available from: http://www.pewinternet.org/2006/10/05/riding-the-waves-of-web-2-0/ [retrieved: April, 2016]

[7] M. B. Kemp, Social computing comes of age, Cambridge, MA: Forrester, 2007.

[8] *FREESIC, Free Secure Interoperable Communication* [Online]. Available from: http://www.freesic.eu. [retrieved: April, 2016]

[9] *NCOIC Interoperability Framework.* [Online]. Available from: http://www.ncoic.org/technology/technical-products/frameworks/10-technology/33-tech-prod-framework-nif [retrieved: April, 2016]

REDIRNET, Emergency Responder Data Interoperability Network. [Online]. Available from: http://www.redirnet.eu. [retrieved: April, 2016]

Towards Drone-Assisted Large-Scale Disaster Response and Recovery

Michael Spranger, Florian Heinke, Sven Becker and Dirk Labudde
University of Applied Sciences Mittweida
Mittweida, Germany
Email: {*name.surname*}@hs-mittweida.de

***Abstract*—Major damaging events with many victims or environmental contamination hazards, such as natural disasters, airplane crashes, train accidents or terroristic acts, are shocking events to society. Fast and comprehensive information acquisition of event sites ensures appropriate and safe actions for rescue forces and investigators, and increases resilience of our society with respect to such disorders in the long term. The use of unmanned aerial vehicles, so-called drones, for gathering as much information as possible about the event site to support the whole resilience cycle is a fast and safe way to elucidate such unknown environments. Therefore, an application framework is currently developed by the authors, which aims at supporting decision makers with respect to targeted and safe management of rescue teams and the fast locating of victims, as well as 3D spatiotemporal modeling and simulation of events for forensic purposes. In this work, we present a process chain for 3D reconstruction of event sites using aerial photogrammetry and open source software.**

Keywords–forensic; unmanned aerial vehicle; resilience engineering; open source; 3D reconstruction

I. Introduction

On March the 24th 2015, *Germanwings* flight 4U9525 crashed in the French Alps after a suicidal co-pilot intentionally initiated a controlled ten minutes lasting descent. All 150 people were killed instantly. Located in remote mountainous terrain difficult to access by vehicles, the disaster site is quickly reachable only by helicopter which posed major logistic problems to rescue forces and investigators, especially in the first hours after the crash. Many other similar scenarios, such as natural disasters, pile-ups, train accidents or terroristic acts, pose analogous problems to personnel, whereas locating survivors and bodies, obstructive and dangerous pieces of debris, and sources of toxic or hazardous chemicals is a major concern in this respect. Thus it is of major importance to generate a general picture based on available information about the disaster site–not only with regards to lifesaving, but strategic resource planning as well. However, especially in the first minutes and hours, such information is difficult to obtain [1] [2].

In this work, we present the concepts of a general framework for drone-assisted 3D reconstruction and mapping of disaster sites. Drones (or unmanned aerial vehicles, UAVs) are small and effective platforms for accessing and imaging locations that are difficult or impossible to reach quickly by personnel or helicopters, and are thus of great potential in providing a fast mapping of disaster sites, as well as locations of potential survivors [1] [3]. Additional drone payloads, such as thermal imaging systems, pollutant sensors, and automated GPS-assisted navigation systems, yield an even wider range of applicability and potential in resilience engineering in general (please see the work of Colomina and Molina [4], as well as Horsman [5] for an extensive overview).

Based on a study presented by Püschel et al. [6], which aimed at 3D reconstruction of tourism objects by combining aerial and terrestrial photogrammetry using drones, we designed concepts for disaster site 3D reconstruction and mapping by means of open source software based on aerial images acquired by drones. In this paper, we first provide background and motivation for said reconstruction processes (see Section II), present the photogrammetry software in question and introduce a strategy for drone-assisted 3D disaster site reconstruction, including results obtained by proof-of-concept testing of such. Finally, in Section III, we briefly introduce our drone system

ISBN: 978-1-61208-481-7

and its conceptual application to the crash site of Flight 4U9525, demonstrate the proposed strategy's performance in application to the Mittweida water tower which we chose as our object of initial studies, and conclude by prospects of future work (Section IV).

II. 3D Disaster Site Reconstruction

An important task in forensic disaster response and recovery is to gather and process information about event sites (such as crime or disaster scenes) quickly and safely in order to support action planning and well-directed employment of rescue teams on the one hand, and eventually provide data for eventual spatiotemporal reconstruction of such an event on the other. There are various methods for gaining such valuable information, including terrestrial and helicopter- or drone-based large-scale imaging, videotaping or laser scanning. The evaluation of the data obtained by mentioned means and subsequent 3D reconstruction can be realized by utilizing various software packages, either licensed, free or open-source. Although cutting edge in quality and performance, laser scanning systems are costly with respect to operation, maintenance and acquisition (including specialized software for data processing). In contrast, open source software can pose cost efficient alternatives. We here propose a pipeline of open source software for 3D event site reconstruction based on drone-assisted imaging.

The first open source software package within the reconstruction process is Visual Structure from Motion (**VisualSFM**) [7]. Providing both a user interface, as well as command line access for batch integration, VisualSFM is utilized for calculating 3D point clouds of multi-image photographs of an object or area based on Wu's scale-invariant feature transform GPU algorithm [8]. In addition there are further algorithms implemented in VisualSFM, such as Clustering Views for Multi-view Stereo (CMVS) and Patch-based Multi-view Stereo (PMVS) [9] to cluster and condense calculated point clouds. In the process, **CMPMVS** [10] is utilized to refine obtained 3D point clouds, reconstruct object surfaces and compute object textures. The downstream software to VisualSFM and CMPMVS are **MeshLab** [11] and **Blender** [12]. MeshLab can be used for post-processing and editing reconstructed surfaces or refining them from computed point clouds if necessary. Furthermore, multiple object meshes can be aligned and unified to one single mesh. Blender is a 3D graphics and animation software and can be used to import object (.obj) files generated in MeshLab and edit, add, merge, measure, (re-)texture and render 3D objects.

Using this pipeline, valid reconstruction results can only be obtained if underlying images are of good quality and, equally important, coherent in perspective. More precisely, images are required to capture the entire scene from all general view angles including perspective overlaps that ensure determining virtual camera positions in the reconstructed 3D point space and, hence, proper object reconstruction. Therefore, analogous to results presented in [13], drone-assisted 3D site reconstruction is of best quality using this software pipeline if the area in question is captured at a circular flight path from a drone circumnavigating the area. Note that computation time and memory usage are determined by the surface reconstruction process, growing exponentially by the number of images, image resolution, and identified object points.

Initially, we tested the concept using sequences of pseudo-aerial images obtained from **Google Earth**. Images were retrieved along virtual circular flight paths and eventually used for 3D site reconstruction. A schematic of the proposed reconstruction workflow is shown in Figure 1 including an example reconstruction of the *Germanwings* Flight 4U9525 crash site based on 25 pseudo-aerial images [14].

III. Drone-based Image Acquisition

A. The Drone in Use: Technical Aspects and Capabilities

The drone used in our study is a *MikroKopter MK-ARF Okto XL 6S12*, an eight-blade rotary wing drone for multi-purpose utilization. In our set-up the MK-ARF Okto XL 6S12 has a maximum slant range of 4,000 m and a maximum ceiling of 5,000 m above sea level. With fully charged batteries and optimal weather conditions, the drone achieves a maximum flight time of about 45 minutes. Besides present weather conditions, maximum flight time is reduced by hardware additionally mounted to the drone, such as a fixed SLR (single lens reflex) camera. Furthermore, the drone is equipped with a CMOS (complementary metal-oxide-semiconductor) camera whose video feed can be received and post-processed on the ground. Although not-movable around the yaw axes, both the CMOS and SLR camera mount can be pitched and rolled. In combination with automated pre-planned waypoint flight and point-of-interest focusing capabilities, as well as automated camera triggering, the user is thus able to obtain images made in-flight at pre-planned positions and altitudes. Waypoints and trigger events can be set-up and uploaded to the drone using the maintenance and control software MikroKopter-Tool V2.12a.

B. A Strategy for Drone-assisted Disaster Site Imaging

As discussed in Section II, 3D object reconstruction by means of VisualSFM using a sequence of images is only feasible if these are recorded along a circular track around the object. Hence, drone-assisted terrain and disaster site reconstruction utilizing this reconstruction strategy requires an analogous object-camera geometry in the recording process. Therefore, a drone is programmed to fly a nearly circular path around the center of the region of interest, whereas the center is constantly focused by the camera. The appropriate image sequence can eventually be obtained in-flight. As elucidated above, the MK Okto XL 6S12 is capable to realize such a pre-planned flight profile.

With respect to extensive dimensions of some disaster sites (i. e., the crash site of Flight 4U9525 is about 380 m $\times$ 500 m), covering the entire region of interest utilizing a single circular path is unfeasible or even impossible to achieve due to range and flight time limitations. To circumvent these restrictions, a given region of interest can be split in a set of smaller overlapping circles in a straightforward manner (see Figure 2A and B). Obtained image sequences are used for 3D reconstruction of corresponding smaller circular areas which are eventually assembled to a single unified model of the region of interest using MeshLab (Figure 2C and D). Image capturing and site reconstruction can thus also be conducted in parallel. In addition, Figure 2E shows a section of the MikroKopter-Tool's waypoint flight planner employed to the Flight 4U9525 crash site. In this virtual scenario, the drone is programmed to take off at a clearing (waypoint two, P2) which can be reached either by helicopter or by vehicle via an unpaved mountain trail and proceeds around the POI (waypoint one, P1) in a circular manner described by 25 waypoints, whereas the camera is pointed towards the POI throughout the flight. Blue circles visualize three additional circular flight paths required for reconstructing almost the entire crash site from corresponding 3D models.

C. Application

The proposed 3D reconstruction strategy was tested on the Mittweida water tower and its close surroundings. The tower is 38 m high and consists of two sections with varying widths of about 10 and 16 m. Featuring large dimensions, small details and free space in its surroundings, it poses a suitable object to test reconstruction capabilities in combination with varying camera-object geometries, camera settings and image resolutions. In Figure 3A, a model of the tower and its surrounding area is shown. Here, the drone was programmed to fly a circle with 50 m radius at an altitude of 50 m above ground level. Images were extracted from recorded HD video material every single second, resulting to 111 images, and processed as proposed. Although major details are discernible, smaller features with a size of about less than one meter are difficult to identify. On a standard desktop machine (eight 3.5 GHz CPUs, 32 GB RAM, GeForce 750 GTX Titan), the 3D reconstruction process required about 1.5 to 2 hours of computation time and 1.5 GB of disk space. Hence, with this set-up, obtained models are only suitable for fast mapping of larger areas. Interestingly, a model computed from only 28 images (four seconds per image) shows only minor discrepancies in quality compared to the model computed from 111 images. Said model is shown in Figure 3B. In addition, computation time for this model is only 20 minutes.

In the reconstruction process of the tower basis the drone was programmed to fly at three meters above ground level with a distance of ten meters from the tower. The obtained model is shown in

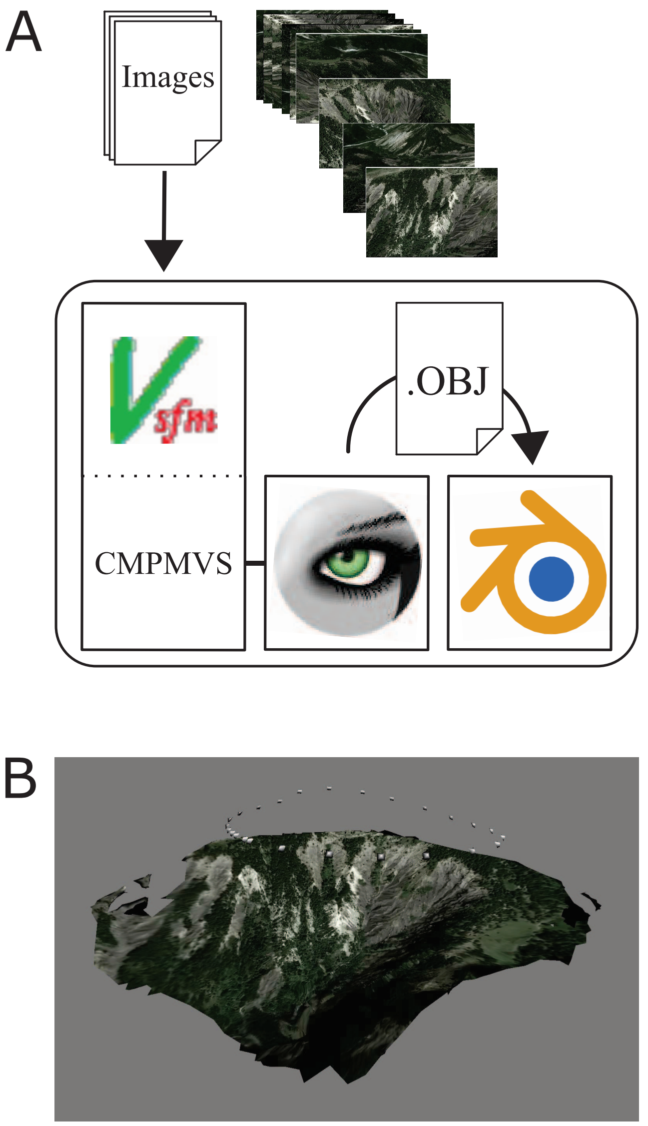

Figure 1. A: Schematic workflow of the proposed 3D disaster site reconstruction strategy using aerial images by means of open source software. Obtained images are input to VisualSFM [7], whereas 3D points for initial object/area reconstruction are computed. CMPMVS [10] is utilized next for point enrichment and mesh construction which is eventually refined or merged with other meshes using MeshLab [11]. Blender [12] is utilized for further refinement and post-processing. Best reconstruction results are obtained if images are recorded on a circular flight path around the object/area of interest. B: 3D model reconstructed from pseudo-aerial images of Flight 4U9525 crash site retrieved from Google Earth [14]. White dots indicate virtual camera locations computed by VisualSFM.

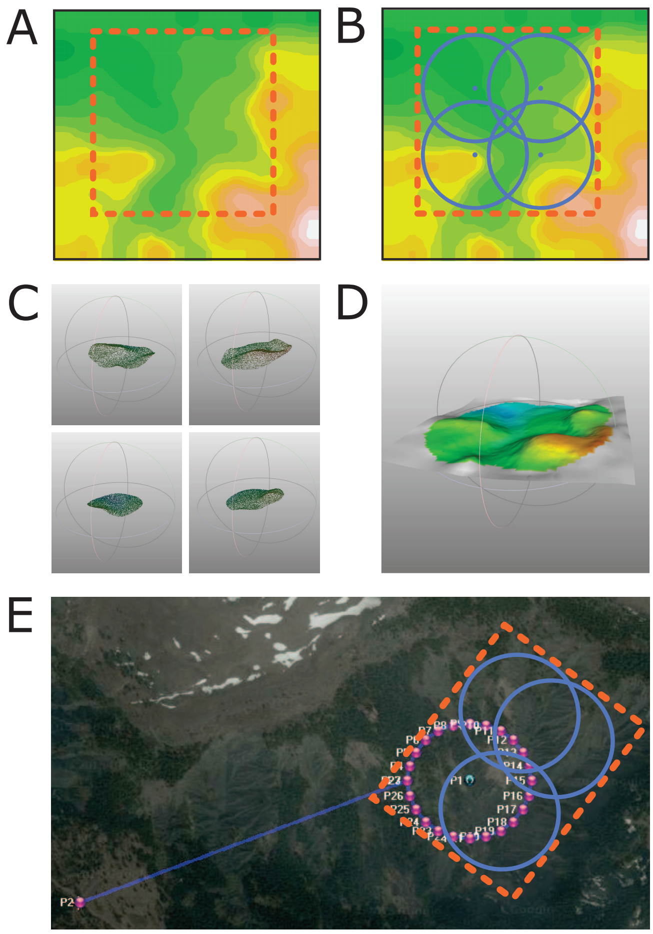

Figure 2. 3D reconstruction strategy for extensive disaster sites. A and B: 3D-reconstruction of a large disaster site is realized by merging computed meshes obtained from image sequences of shorter circular flights covering the entire area. C and D: Overlapping flight paths ensure proper mesh alignment and unification. E: Screen capture of a planned circular flight in MikroKopter-Tool V2.12a shown for the Flight 4U9525 crash site [14] as the region of interest. Throughout the flight, the camera is pointed towards the point of interest (P1). Three additional overlapping circular flights (indicated by blue circles) are required to reconstruct the area based on obtained image sequences.

Figure 3C. Here, sixty 24 megapixel images were recorded at an angular offset of 6° using interval triggering. Time and disk space demands are significantly larger for this set-up (6.5-8 hours, 53 GB disk space). Computational demands are accompanied with a high degree in object and surface detail, even for small objects ($<$ 10 cm), suitable for detailed reconstruction and mapping of smaller areas.

In summary, camera-object geometry, image resolution and the number of considered images has to be chosen in accordance to the features of the target object/area and the problem to address. Especially computational time demands have to be taken into account during flight planing.

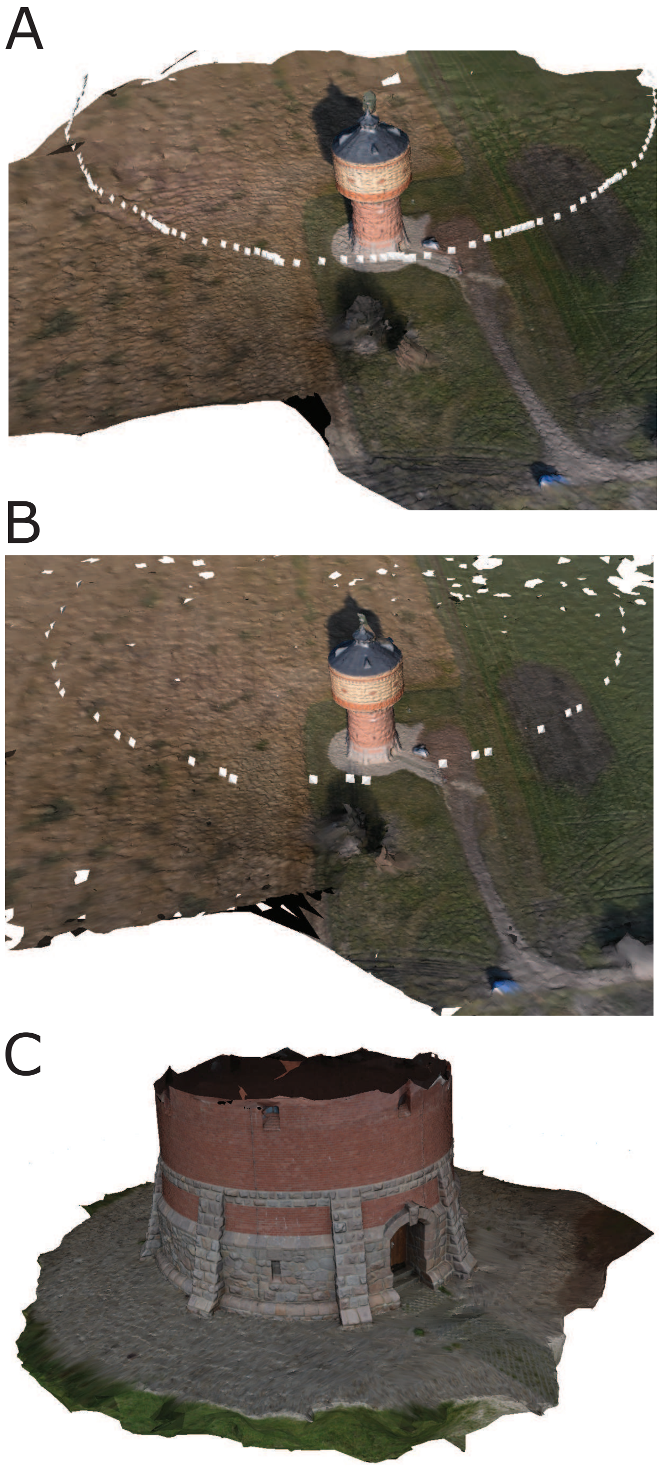

Figure 3. A and B: 3D model of the Mittweida water tower (height: 38 m) obtained from 111 respectively 28 images extracted from HD video. C: 3D model of the tower basis (width: 10 m) generated from sixty 24 megapixel images.

IV. Conclusion and Future Work

Fast reconstruction of disaster sites can be of great value in disaster response and recovery. The presented work is focused on the conceptual design of such a 3D disaster site reconstruction strategy by means of open-source software based on aerial images obtained by drones. Models generated from images obtained during flight show that model quality is greatly dependent on camera-to-POI geometry and image resolution. It is further pointed out that even the resolution of images extracted from high definition video is not always sufficient for detailed disaster scene reconstruction. Although lower image resolutions lead to coarse models, time demands for automated model calculation are relatively small and obtained level of detail can be sufficient for fast mapping processes, which makes these models adequate for the exploration of large disaster sites and providing support to the rescue forces in response planning. High resolution models are achieved by using high resolution images (e. g., 24 megapixels), whereas computational demands increase significantly.

Future work requires the recording of more aerial drone-based images and evaluation of generated meshes, whereas the conceptual strengths and weaknesses are ought to be identified and verified. Here, the focus lies on the quantification of reconstruction error. Furthermore, specialized payloads such as thermal imaging systems shall be considered in the future.

References

[1] H. Bendea et al., “Low cost UAV for post-disaster assessment,” The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 37, no. Part B, 2008, pp. 1373–1379.

[2] Y. Naidoo, R. Stopforth, and G. Bright, “Development of an UAV for search & rescue applications,” in AFRICON, 2011. IEEE, 2011, pp. 1–6.

[3] L. Barazzetti et al., “3D scanning and imaging for quick documentation of crime and accident scenes,” in SPIE Defense, Security, and Sensing. International Society for Optics and Photonics, 2012, pp. 835 910–835 910.

[4] I. Colomina and P. Molina, “Unmanned aerial systems for photogrammetry and remote sensing: A review,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 92, 2014, pp. 79–97.

[5] G. Horsman, “Unmanned aerial vehicles: A preliminary analysis of forensic challenges,” Digital Investigation, vol. 16, 2016, pp. 1–11.

[6] H. Püschel, M. Sauerbier, and H. Eisenbeiss, “A 3D model of Castle Landenberg (CH) from combined photogrammetric processing of terrestrial and UAV-based images,” Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci, vol. 37, 2008, pp. 93–98.

[7] M. Ziegler, E. Gülch, and P. Rawiel, “3D-Rekonstruktion von Objekten mittels Structure-from-Motion aus einer photogrammetrischen Aufnahme mit den Programmen VisualSFM und CMPMVS,” in DGPF Tagungsband, vol. 23.

[8] C. Wu, “SiftGPU Implementation of Scale Invariant Feature Transform.” [Online]. Available: http://cs.unc.edu/ ccwu/siftgpu/, retrieved: April 8, 2016

[9] Y. Furukawa and J. Ponce, “Accurate, Dense, and Robust Multiview Stereopsis,” IEEE Transactions on Pattern Analysis and Machine Intelligence,, vol. 32, no. 8, Aug 2010, pp. 1362–1376.

[10] M. Jancosek and T. Pajdla, “Multi-view reconstruction preserving weakly-supported surfaces,” in 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 3121–3128.

[11] MeshLab. [Online]. Available: http://meshlab.sourceforge.net/, retrieved: April 8, 2016

[12] Blender. [Online]. Available: https://www.blender.org/, retrieved: April 8, 2016

[13] P. L. Falkingham, “Acquisition of high resolution three-dimensional models using free, open-source, photogrammetric software,” Palaeontologia Electronica, vol. 15, no. 1, 2012, p. 15.

[14] Google Earth, retrieved: January 19, 2016, ’Germanwings Flight 4U9525 crash site’ 44°16’48.46” N 6°26’18.84” E, Image recording date: March 29, 2015.

In-Vehicle Interface Assessment Framwork for Emerging Technologies

A case study for evaluating new driver-vehicle interaction design

Myoungouk Park, Bawul Kim, Joonwoo Son*
Convergence Research Center for Future Automotive Technology
Daegu Gyeongbuk Institute of Science and Technology (DGIST)
Daegu, South Korea
Emails: violet1211@dgist.ac.kr, kimx2277@dgist.ac.kr, json@dgist.ac.kr

***Abstract*—This paper proposes an in-vehicle interface assessment framework for emerging technologies. The framework was validated through a driving simulator based case study with an emerging user interface design. The result suggested that it was useful to evaluate the effectiveness of the emerging technology based user interface designs.**

Keywords-Usability Assessment; Driver-Vehicle Interaction (DVI); User Interface; Emerging Technology.

I. INTRODUCTION

With advancing technology, drivers are more likely to engage in non-driving related tasks. Auto industries are attempting to develop new user interaction designs, such as voice and gesture command to reduce drivers' distraction [1]. However, it is important to determine if new concept of driver-vehicle interaction design is indeed effective.

Recent research suggested a usability evaluation toolkit for In-Vehicle Information Systems (IVISs) [2]. The toolkit comprises definition of usability criteria, selection of evaluation methods, desktop methods, and experimental methods. However, detailed information of the experimental methods was limited.

This paper aims to suggest and validate an in-vehicle interface assessment framework based on experimental methods.

II. IN-VEHICLE INTERFACE ASSESSMENT FRAMEWORK

A proposed assessment framework is illustrated in Figure 1. The framework begins with definition of usability criteria, which definition had to be context specific. Then usability criteria are used to guide the selection of methods which are most appropriate for evaluating usability. In the experimental methods phase, objective and subjective evaluations should be repeated until the usability criteria are met.

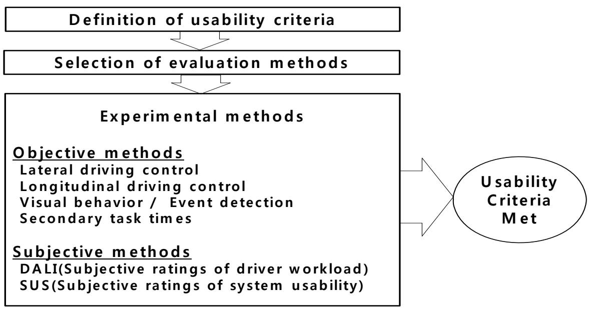

Figure 1. In-vehicle interface assessment framework.

III. CASE STUDY

A. Study Overview

A quantitative assessment framework was developed to understand the effectiveness of an emerging user interaction device on distraction. A comparative analysis between new user interface (NUI) and touch screen (TS) was conducted.

B. Definition of Usability Criteria

The usability criteria of new in-vehicle interface was defined by identifying input/output modalities and the context of use as shown in Table I and Table II.

C. Selection of Evaluation Methods

Objective and subjective methods were selected to evaluate actual performance levels and users' opinions as shown in Table III. Two subjective methods, i.e., the System Usability Scale (SUS) [3] and the Driving Activity Load Index (DALI) [4], were selected. The driving performance (primary task), secondary task and eye behavior were selected as the objective measures.

D. Experimental Setup

The simulator experiment was conducted in a fixed-based driving simulator. A gaze tracker was mounted on a dash board to collect eye behavior data. A touch pad for new input method were placed beside a gear lever.

TABLE I. INPUT AND OUTPUT DEFINITION

System	Input	Output
New User Interface	Touch Pad	Visual & Auditory
Conventional Interface	Touch Screen	Visual & Auditory

TABLE II. CONTEXT OF USE AND USABILITY CRITERIA

Factors	Criteria	Experiment Design
Dual Task Environment	Effectiveness, Efficiency, Interference	Secondary Tasks: AUI & Touch Screen
Environmental Condition	Effectiveness under varying driving conditions	Simulated Road Env.: Highway & Rural
Training Provision	Learnability	Training & Practice

 ISBN: 978-1-61208-481-7

TABLE III. SUBJECTIVE AND OBJECTIVE MEASURES

Domains	Measures	Method	Tool
Subjective Evaluation	Usability	Survey	SUS
	Workload	Survey	DALI
Driving Performance	Velocity	Simulated driving	Driving Simulator
	Steering Reversal Rate		
	Standard deviation of Lane Position		
	Lateral Acceleration		
	Lane crossing		
Eye Behavior	Single Glance Time	Eye tracker	FaceLAB 4.6
	Total Glance Time		
	Percent glance durations≥ 1.6s		
	Number of Glance		

E. Experimental Procedure

Twenty four participants, consisted of 12 driving group (Driving & Survey) and 12 non-driving group (Survey only), were recruited. Following informed consent and completion of a questionnaire, participants received 10 minutes of adaptation time in the simulator. Then, participants were trained in the NUI and TS operation. When the simulation was resumed, participants drove on a highway for about 20 minutes twice to perform either the NUI or the TS task. The tasks consisted of destination entry, MP3 play, emergency mode, and mute function.

F. Results

As shown in Figure 2, overall SUS results showed that the NUI score (65.7) was 6.7 percent higher than the TS score (61.1). Especially, the driving and survey group have rated significantly higher than the survey only group (p=0.023). Among the six items in DALI, five items of the NUI, including global attention demand, visual demand, stress, temporal demand, and interference, showed significantly lower workload than the touch screen (see Table IV).

For the objective methods, eye movement and driving performance changes are summarized in Table V. In general, the NUI showed higher performance and safer behavior than the touch screen.

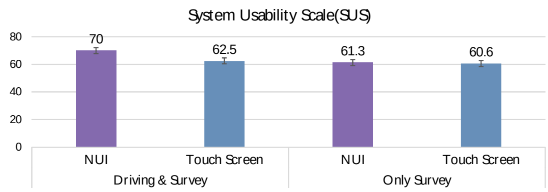

Figure 2. Result of System Usability Scale (SUS)

TABLE IV. SUBJECTIVE AND OBJECTIVE MEASURES

Questions	Driving & Survey		Only Survey	
	NUI	*Touch Screen*	*NUI*	*Touch Screen*
Global attention demand*	2.62	3.69	2.58	3.25
Visual demand**	2.46	3.85	2.42	3.25
Auditory demand	2.54	2.69	1.58	1.83
Stress**	2.23	3.15	2.17	3.25
Temporal demand**	2.15	3.54	1.75	2.92
interference**	2.31	3.69	2.08	3.25

Note: Device type (* p < .05 ** p < .01)

IV. CONCLUSION

This study proposed an in-vehicle interface assessment framework for emerging technologies which have not been used in automotive user interaction design. The results of the case study have shown that the proposed framework have suitable levels of validity. It was also demonstrated that the experimental methods using a driving simulator were useful to evaluate the effectiveness of the emerging technology based user interface designs which are hard to imagine their use cases in a driving context.

ACKNOWLEDGMENT

This research was supported in part by DGIST (Daegu Gyeongbuk Institute of Science and Technology) Research Program (Project No. 16-IT-02).

REFERENCES

[1] C. A. Pickering, K. J. Burnham, and M. J. Richardson, "A review of automotive human machine interface technologies and techniques to reduce driver distraction", Proc. The 2nd IET International Conference on System Safety, IEEE Press, Oct. 2007, pp. 223-228.

[2] C. Harvey, N. A. Stanton, C.A. Pickering, C. A., M. McDonald, and P. Zheng,. "A usability evaluation toolkit for in-vehicle information systems (IVISs)", Applied ergonomics, vol. 42, no. 4, 2011, pp. 563-574.

[3] J. Brooke, "SUS: A quick and dirty usability scale", Usability evaluation in industry, London: Taylor & Francis, vol. 4, no. 7, 1996, pp. 189 -194.

[4] A. Pauzié, "Evaluating driver mental workload using the driving activity load index (DALI)", In Proc. of European Conference on Human Interface Design for Intelligent Transport Systems, April 2008, pp. 67-77.

TABLE V. DRIVING PERFORMANCE AND EYE BEHAVIOR MEASURES

		New UI				Touch Screen			
		Destination entry	*MP3 play*	*Emergency mode*	*Mute function*	*Destination entry*	*MP3 play*	*Emergency mode*	*Mute function*
Driving Performance	Velocity	90.85	91.75	90.25	95.78	93.94	88.77	90.98	89.96
	SRR	7.30	6.32	6.02	3.36	8.36	10.44	6.17	7.63
	SDLP	0.29	0.29	0.27	0.18	0.37	0.39	0.19	0.39
	Lateral Acceleration	2.36	2.43	2.56	2.46	2.54	2.69	2.52	2.41
	Lane crossing	0.31	0.00	0.00	0.28	0.65	0.70	0.00	0.56
Eye Behavior	Single Glance Time	1.24	0.91	0.58	0.35	1.29	1.13	0.69	1.16
	Total Glance Time	10.02	7.83	1.22	0.56	9.79	10.2	1.56	1.84
	Percent glance durations ≥1.6s	1.69	0.85	0.15	0.00	1.46	1.92	0.15	0.39
	Number of Glance	8.23	8.54	1.00	0.85	8.15	9.39	2.15	1.62

Real-Time Detection and Classification of Driver Distraction Using Lateral Control Performance

Joonwoo Son
Convergence Research Center for Future Automotive Technology
Daegu Gyeongbuk Institute of Sci. & Tech. (DGIST)
Daegu, Republic of Korea
json@dgist.ac.kr

Myoungouk Park
Convergence Research Center for Future Automotive Technology
Daegu Gyeongbuk Institute of Sci. & Tech. (DGIST)
Daegu, Republic of Korea
violet1211@dgist.ac.kr

***Abstract*— This paper suggests a real-time method for detecting both visual and cognitive distraction using lateral control performance measures including standard deviation of lane position (SDLP) and steering wheel reversal rate (SRR). The proposed method adopts neural networks to construct detection models. Data for training and testing the models were collected in a driving simulator in which fifteen participants drove through a highway. They were asked to complete either visual tasks or cognitive tasks while driving to create distracted driving periods. As a result, the best performing model could detect distraction with an average accuracy of 93.1%.**

Keywords-driver distraction; distraction classsification; driving performance; machine learning; neural network.

I. INTRODUCTION

Recent technological advances have enabled a wide variety of information systems to be integrated into a vehicle in order to increase safety, productivity, and comfort. However, drivers are also exposed to more distraction sources than before [1]. The driver's distraction is a specific type of inattention that occurs when drivers divert their attention away from the driving task to focus on another activity instead [2]. The major types of in-vehicle distraction can be categorized into visual-manual and cognitive distraction.

There have been efforts to monitoring driver's distraction in real time using driving performance [3], eye movement measures [4][5], and physiological measures [6]. However, most previous studies have focused on a specific distraction type, either visual or cognitive.

Thus, this paper presents results using neural networks for detecting and classifying visual and cognitive driver's distractions trained using lateral control performance measures, including the Standard Deviation of Lane Position (SDLP) and Steering wheel Reversal Rate (SRR).

II. DISTRACTION CLASSIFICATION MODEL

As shown in Figure 1, lateral performance measures including SDLP and SRR have different profiles according the type of distraction. Based on this behavioral difference, the distraction detection and classification model was constructed.

A. *Experimental Setup for Learning Data Collection*

The experiment was conducted in the DGIST fixed-based driving simulator, which incorporated STISIM Drive™ software and a fixed car cab. The virtual roadway was displayed on a wall-mounted. Sensory feedback to the driver was also provided through auditory and kinetic channels. Distance, speed, steering, throttle, and braking inputs were captured at a nominal sampling rate of 30 Hz.

B. *Generation of distraction*

In this study, visual distraction was generated by an arrow search task, which only required visual processing demand and minimal cognitive processing [7]. The arrow search task had three different arrangements of arrows to create three levels of difficulty. Cognitive distraction at three distinct levels was created using an auditory delayed digit recall task, so called n-back task. The n-back task requires participants to repeat the nth stimulus back in a sequence [8].

C. *Experimental Procedure*

Fifteen young males, in the 25-35 age range (M=27.9, SD=3.13), were recruited to collect visually and cognitively distracted driving data. Following informed consent and completion of a pre-experimental questionnaire, participants received 10 minutes of adaptation time in a simulator. The simulation was then stopped and participants were trained in the n-back task while remaining seated in the vehicle. When the simulation was resumed, participants drove on a straight highway twice, one for visual distraction and the other for cognitive. Each driving takes about 20 minutes, and participants perform a secondary task, i.e., n-back task or arrow task at a specified segment.

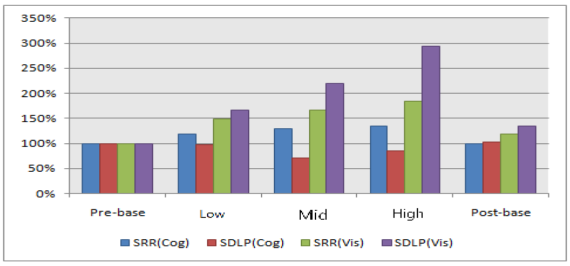

Figure 1. Comparison of lateral control performance

D. Input Features

The SDLP and the SRR were used for detecting both types of distraction in the classification models. The SRR was calculated by counting the number of steering wheel reversal from the low pass filtered steering wheel angle per minute. For cognitive distraction, the selected cut-off frequency of the low pass filter was 2Hz and the gap size of the reversal angles was 0.1 degree. For visual distraction, the cut-off frequency and the gap size were 0.6Hz and 3 degrees. The SDLP in both distraction types was calculated from 0.1Hz high pass filtered lateral position based on the AIDE report [9].

E. Model Training and Testing

Radial Basis Probabilistic Neural Networks (RBPNN), which are known as one of suitable methods for classification problems [10], were used to construct the driver's distraction classification model. For training and testing the distraction detection models, the simulated driving data sets were used. Each data set consists of a driving only period and three levels of distracted driving periods. Each task duration was divided into multiple segments based on a time window size. This study considered seven window sizes, i.e., 2, 3, 5, 10, 15, 20 and 30 seconds. Among the segments in each task, half of them were used for training and the others for testing.

III. Result And Discussion

The performance of the distraction detection models varies depending on the window sizes. As shown in Table 1, the time windows between 3 and 10 seconds provided good performance in overall. Under the visual distraction, the highest accuracy was 98.5% with 10 seconds window, but the model performance was degraded when the window sizes are smaller than 3 seconds or bigger than 15 seconds. In the cognitive distraction, the best accuracy was 93.6% with 2 seconds window.

In general, the SRR represents the control effort needed to cope with time sharing induced by a secondary task, and thus provides a direct measure of the consequences of visual or cognitive demand on lateral control. Thus, the increased SRR could be interpreted in terms of increased workload. Regarding the SDLP, the increased SDLP is often observed under visual distraction, but cognitive distraction causes the reduced SDLP [11]. Due to the characteristics of lateral performance measure, the classification performance in visual distraction could have specific regions of window size to provide better accuracy rate.

TABLE I. Model Performance With Different Window Sizes

Size (sec.)	Total Accu-racy	Cognitive Accuracy				Visual Accuracy			
		Low	*Mid*	*High*	*Avg.*	*Low*	*Mid*	*High*	*Avg.*
2	86.4	91.6	94.9	94.2	93.6	83.3	79.1	75.3	79.3
3	90.0	83.7	86.7	88.7	86.3	96.0	93.3	91.7	93.7
5	93.1	86.7	90.0	94.4	90.4	97.8	96.1	93.3	95.7
10	92.8	80.0	88.9	92.2	87.0	100.0	96.7	98.9	98.5
15	85.8	90.0	88.3	100.0	92.8	71.7	80.0	85.0	78.9
20	83.7	80.0	77.8	95.6	84.4	75.6	84.4	88.9	83.0
30	81.7	73.3	80.0	83.3	78.9	83.3	86.7	83.3	84.4

IV. Conclusion

In this paper, we proposed a real-time method for detecting both types of driver's distraction using the lateral control performance measures including SDLP and SRR. In order to collect training and testing data, fifteen participants drove in a driving simulator and completed three different levels of cognitive and visual tasks. The distraction detection and classification was performed by RBPNN models.

The results show that the proposed models were able to detect both types of driving distraction with high accuracy. The model performance was assessed with the cross-validation scheme. As a result, the highest accuracy rate in overall model performance was 93.1%. And it is also expected that the accuracy can be improved by applying more sophisticated algorithms and supplementary inputs.

Acknowledgment

This research was supported in part by DGIST Research Program (Project No. 16-IT-02) and Global Research Laboratory Program (Project No. 2013K1A1A2A02078326) of the Ministry of Science, ICT and Future Planning (MSIP).

References

[1] B. Donmez and Z. Liu, “Associations of distraction involvement and age with driver injury severities,” J. of safety research, vol. 52, Feb. 2015, pp. 23-28.

[2] T.A. Ranney, “Driver distraction: A review of the current state-of-knowledge,” No. DOT HS 810 704, Washington, DC: National Highway Traffic Safety Administration, 2008.

[3] Y. Zhang, Y. Owechko, and J. Zhang, “Driver cognitive workload estimation: A data-driven perspective,” Proc. the 7th International IEEE Conference on Intelligent Transportation Systems (ITSC 2004), IEEE press, Oct. 2004, pp.642-647.

[4] L. M. Bergasa, J. Nuevo, M. Sotelo, R. Barea, and M.E. Lopez, “Real-time system for monitoring driver vigilance,”. IEEE Trans. Intelligent Transportation Systems, vol. 7, no. 1, March 2006, pp. 63-77.

[5] Y. Liang, M. L. Reyes, and J. D. Lee, “Real-time detection of driver cognitive distraction using support vector machines,” IEEE Trans. Intelligent Transportation Systems, vol. 8, no. 2, June 2007, pp. 340-350.

[6] J. Healey and R. W. Picard, “Detecting stress during real-world driving tasks using physiological sensors,” IEEE Trans. Intelligent Transportation Systems, vol. 6, no. 2, June 2005, pp. 156-166.

[7] J. Engström, E. Johansson, and J. Östlund, “Effects of visual and cognitive load in real and simulated motorway driving,” Transportation Research Part F: Traffic Psychology and Behaviour., vol. 8, no. 2, March 2005, pp. 97-120.

[8] B. Mehler, B. Reimer, and J. A. Dusek, “MIT AgeLab delayed digit recall task (n-back),” Cambridge, MA: Massachusetts Institute of Technology, 2011.

[9] J. Östlund, B. Peters, B. Thorslund, J. Engström, G. Markkula, A. Keinath, D. Horst, S. Juch, S. Mattes, and U. Foehl, “Driving performance assessment – methods and metrics,” No. IST-1-507674-IP, Information Society Technologies, 2004.

[10] H. Demuth and M. Beale, “Neural Network Toolbox User’s Guide,” MathWorks, Inc., Natick, MA, 2004.

[11] J. Son, Y. Lee, and M. Kim, “Impact of traffic environment and cognitive workload on older drivers’ behavior in simulated driving,” Int’l J. of Precision Engineering and Manufacturing, vol. 12, No. 1, Feb. 2011, pp. 135-141.

Evolution of Automated Regression Testing of Software Systems Through the Graphical User Interface

Pekka Aho
Knowledge Intensive Products and Services
VTT Technical Research Centre of Finland
Oulu, Finland
email: pekka.aho@vtt.fi

Emil Alégroth
Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden
email: emil.alegroth@chalmers.se

Rafael A. P. Oliveira
Dept. of Computer Systems
University of São Paulo
São Carlos, Brazil
email: rpaes@icmc.usp.br

Tanja E. J. Vos
Centro de Mtodos de Produccin de Software (ProS)
Universidad Politecnica de Valencia
Valencia, Spain
email: tvos@pros.upv.es

***Abstract*—Increasing and more ubiquitous use of mobile and Web applications with graphical user interfaces (GUIs) places more stringent requirements on the software's quality. Automated testing is used to ensure the quality but testing through the software's GUI is challenging and therefore a research topic that has grown during the last decade. However, despite of the evolution of automated GUI-based testing methods and tools, its industrial adoption has been limited. In this paper, we present a synthesis of the evolution of GUI-based test automation and propose a classification for methods and tools for automated regression through the GUI.**

Keywords-Graphical user interface; automated GUI testing; software systems; classification; categorization; state-of-the-art.

I. INTRODUCTION

Increasing and more ubiquitous use of all kinds of mobile and Web applications with GUIs makes our daily lives dependent on the software functioning without errors, increasing the importance of assuring the correct and reliable behavior of software systems. Modern GUI-driven applications are often connected and consist of distributed back-end services and sub-systems. Additionally, the GUI is often the primary interface to access the software's functionality, which also makes it a natural interface for testing, and in some cases, the only means to perform end-to-end testing.

The widespread use of iterative and incremental processes and continuous integration practices in software development has shortened release cycles and limited the time available for testing in each release. This trend poses a challenge since manual GUI-based testing is tedious, laborious [1], and requires a lot of time. This implies that GUI-based test automation, especially for regression testing should be applied to get confidence in the quality of each release. However, from the point of view of continuous integration processes, GUI-based testing is often too slow to be run after each code commit, because the test automation tool has to wait for the GUI to react before executing the next action of the test sequence. Larger automated GUI testing suites are therefore run only a few times a day or overnight.

According to IEEE Standard Glossary of Software Engineering Terminology [2], regression testing is "Selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements." As such, regression testing aims to verify that the behavior of the system under test (SUT) remains consistent after changes to the SUT. Thus, if the changes in SUT intentionally affect the SUT's behavior, the regression test cases usually have to be updated to correspond to the new behavior. Otherwise, the changes may have been unintentional and a regression fault was found.

There are various terms used for automated GUI testing or GUI-based testing, depending on the authors and the objectives of the testing. In our case, automated testing of software systems through GUI would be the most accurate, but the other terms are used as well. The main point is that we are not testing only the software related to the GUI, but using the GUI as an interface for testing the whole software, including also the possible back-end services. The approach is usually black-box, without access to the source code of the system and often without detailed knowledge on the architecture or implementation of the system, and system testing focusing on the functional requirements, features and behavior of the system. However, automated GUI-based testing provides opportunities also for non-functional testing, such as performance and robustness testing.

Automated testing of software systems through the GUI is challenging and has therefore become a popular research topic during the last decade. Despite of the evolution of automated GUI-based testing methods and tools, no large scale industrial adoption of state-of-the-art methods and tools has been seen, and capture and replay (C&R) tools remain being the most popular GUI testing approach in the software

 ISBN: 978-1-61208-481-7

industry. However, C&R tools are associated with high maintenance costs implying a need for more cost-effective GUI-based testing.

In this paper, we summarize the evolution of automated testing of software systems through the GUI in Section II and propose a classification for the methods and tools in Section III. We present the related work in Section IV, and Conclusion and Discussion in Section V.

II. EVOLUTION OF AUTOMATED GUI TESTING

C&R, also called as record and replay (R&R), is one of the earliest and most widely used approaches for automating regression testing of GUI software. In C&R approaches, a test automation tool is used to capture the user's interactions with the SUT during manual use. The tool can then automatically replay the recorded sessions or sequences of interactions against different versions of the software, automating the test execution for regression testing. The modern, more advanced C&R tools capture also the behavior of the GUI software and are able to notice if the behavior of a later version changes compared to the recorded behavior. Usually, each test case is a session of manual interaction and has to be recorded separately.

In general, the C&R approaches are easy and intuitive to use, the tools are mature and widely used, and it is possible to get fast results, for example decreasing the manual effort for regression testing through GUI. There is a wide selection of both commercial and open source C&R tools for most of the widely used operating systems (OS) or platforms, such as SeleniumHQ [3] for Web applications, Appium [4] for mobile applications, and Squish [5] for a variety of different platforms, although all of them can be used for more than just C&R testing.

The obvious disadvantages of C&R is the amount of manual effort required to record the test cases, and even more importantly, the amount of manual effort required to maintain the test suites. Hence, whenever the software changes, the test cases related to the changed parts of the GUI have to be manually retested to be recorded again.

The next step in the evolution of automated GUI testing was using keywords and action words to present the GUI testing scripts on a higher level of abstraction. The goal was to make it easier to reuse parts of test cases to create new test cases and reduce the maintenance effort of test suites after changes in the GUI by providing a clear separation of concerns between business logic and the GUI navigation needed to implement the logic [6]. Although one could argue that the modern C&R tools are exploiting keywords to allow easier maintenance of test cases, purely keyword-based approaches for GUI testing have not been widely adopted by the industry.

When model-based testing (MBT) was introduced in the testing community, it was also adopted into automated GUI testing. In model-based GUI testing (MBGT), the GUI and its behavior is modeled in a higher level of abstraction, using a modeling language supported by the selected test generation tool. In traditional MBGT, the models are created manually, and the generated abstract test cases have to be mapped or transformed into a lower level of abstraction to get concrete executable test cases that can be automatically executed against the SUT. In addition to the effort required to create the models for MBGT, also considerable expertise on formal modeling is required. TEMA Toolset [7] is an example of using MBT for testing concurrency issues in smartphone applications through the GUI.

In recent years, model extraction, also called as model inference, specification mining, reverse engineering or GUI ripping, has been widely used to automatically extract GUI models for testing purposes. The earliest approaches used static analysis on the source code of GUI software, which had the drawback that it failed to capture the dynamic behavior of the GUI. In dynamic analysis, the behavior of the GUI is instead analyzed during runtime interaction with the SUT. Some tools using dynamic analysis for model extraction require a user to interact with the GUI, in a similar way to C&R tools, but more recent tools are able to simulate the end user, automatically interacting with the components or widgets of the GUI.

Most dynamic model extraction approaches, such as [8]-[12], use the following process to capture the GUI model: 1) Capture the current state of the GUI as a snapshot of the screen visible for the end user, 2) Update the behavioral model of the GUI if it is extracted, 3) Analyze the interactions that are available for the end user, 4) Select one of the interactions using a random or a more intelligent selection strategy, 5) Execute the selected action and wait for the GUI to update, 6) Repeat the process from step 1. There are small differences on what is considered as a state of the GUI, but usually it consists of the windows or screens visible to the user, the components or widgets of each of the screens, and properties and values of each of the widgets. If a behavioral GUI model is extracted as well, the differences between the approaches are more significant. Although there are a lot of publications around GUITAR [8] that uses event-based models, most approaches use finite state-machine (FSM) –based models and graphs to present the behavior of the GUI.

In most approaches, the GUI state, a snapshot of the visible GUI, is captured using some kind of application programming interface (API) provided by the OS, such as Windows Automation API [13], or a GUI library, such as Jemmy framework [14] for Java-based GUIs. The benefit of these APIs is that they provide the GUI information in a detailed and hierarchical way. The downside is that such APIs have not been standardized and in practice, model extraction tools have to implement support for several APIs and libraries to cover a wide variety of GUIs. Another option is Visual GUI Testing (VGT), using image recognition on partial images of the GUI and screen captures to extract the state of the GUI [15]. The benefit is the independence of the platform specific APIs and libraries, but the downside is that the visual approaches are not as accurate and detailed. In optimal cases, the model extraction tools are able to reach all parts of the GUI and extract an accurate behavioral model of the GUI. However, automatically extracting GUI models is still an active research topic.

The obvious restriction with the extracted models is that they are based on the observed behavior of the

 ISBN: 978-1-61208-481-7

implementation, instead of expected or required behavior defined in requirement specifications. Therefore the extracted models are ill suited for conformance testing without manually elaborating the models before test case generation. However, the extracted models can be used for reference or regression testing. In optimal cases, using extracted models for generating test cases may achieve a high level of automation in regression testing through the GUI [8], but the quality and effectiveness of these approaches maybe lacking for software that is still under development. The challenge is that when the GUI changes, the automatically extracted reference model has to be extracted again, and then the test cases have to be generated based on the new reference model. As a result, the old test cases are failing and giving false positives for the changed parts of the GUI, and the new parts of the GUI are completely missing from the old test cases. Although the process of model extraction, test case generation and test execution can be fully automated, updating the reference model results a GUI version that has to be tested manually or using other means to ensure the correctness of the new reference model. Otherwise the newly generated test cases could use faulty behavior as their test oracle.

The latest step in the evolution of automated regression testing through GUI has been automated regression analysis based on comparison of automatically extracted GUI models [9]. This approach overcomes the problem of having to re-generate the test cases by not having test cases at all. Whenever the GUI changes, a new model is automatically extracted and compared to the previous version. All the changes are reported for the test engineer, and the manual work is limited to deciding if the change was intentional or a regression fault.

In addition, a lot of smaller scale evolution is studied in academia, improving the automated regression testing through the GUI. For example, automating the debugging of failures found during automated GUI testing [16] is definitely improving the level of automation of the whole software development process.

III. CLASSIFICATION FOR AUTOMATED TESTING OF SOFTWARE SYSTEMS THROUGH GUI

In this section, we propose a 2-axis classification of methods and tools for automated testing of software systems through the GUI, illustrated in Figure 1. Our intention is to provide a baseline for comparison between tools and methods for automated GUI-based testing, as a suitable public categorization is currently missing. Many of these methods and tools for GUI-based testing are still academic or proof-of-concept tools, but we hope that in the future, when the tools have matured and there is more tools to select from, this classification helps the industry in selecting the tools and methods suitable for their needs.

The vertical axis of our classification follows the three generation classification proposed by Alégroth et al. [15] but

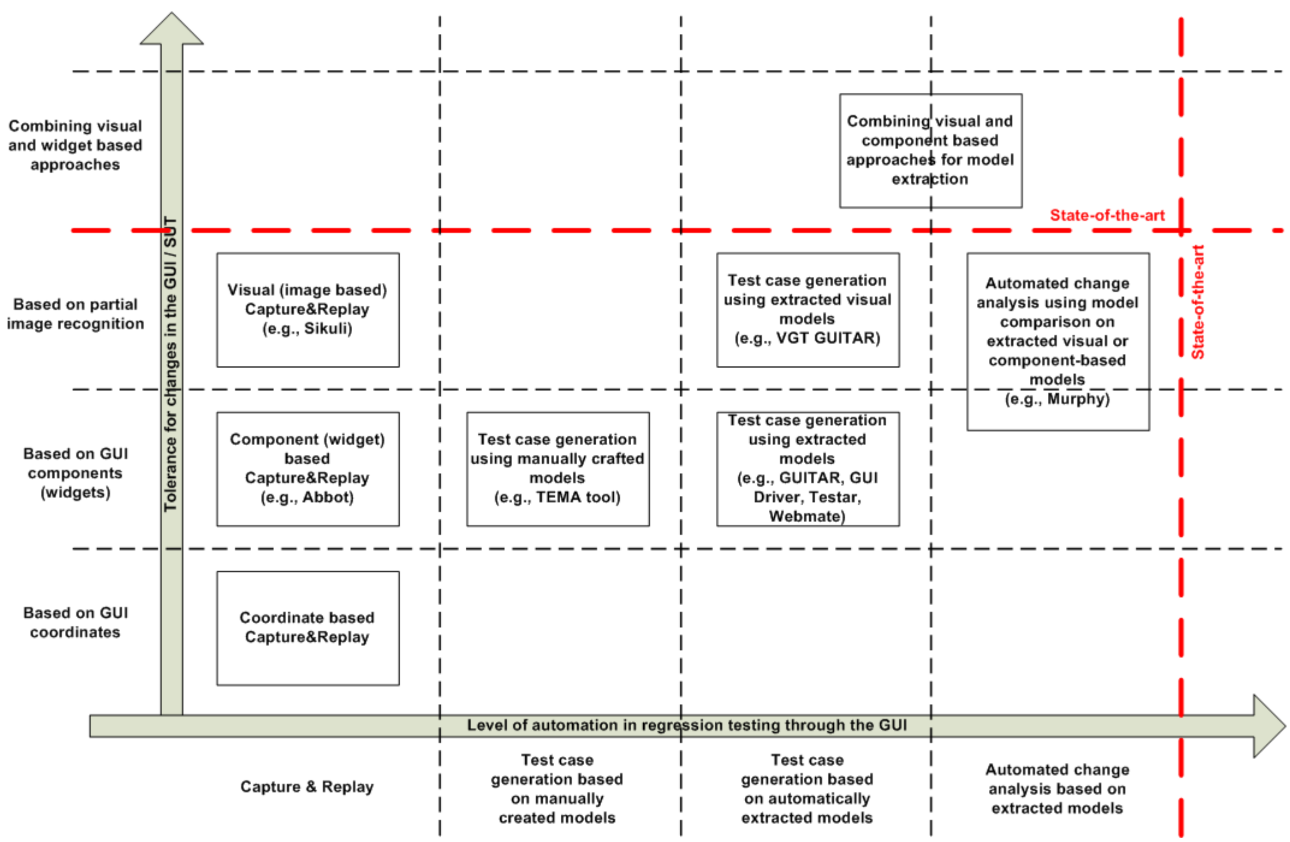

Figure 1. Classification for automated regression testing through GUI.

 ISBN: 978-1-61208-481-7

is named as tolerance for changes in the GUI. We have also included a fourth generation, not discussed by Alégroth et al., that combines the visual and API-based approaches, getting the benefits of both approaches, as discussed in Section II. As the combination would not bring any novel approaches to the field, we could call it 3.5th generation.

The earliest C&R approaches captured user actions in exact mouse coordinates [15]. These coordinate-based tools are categorized into lowest level of tolerance for changes in the GUI, because even small changes in the GUI, such as changing screen resolution or window's location on the screen, usually breaks the test case. None of the modern tools are relying solely on mouse coordinates anymore.

The second generation [15], or the second level of tolerance, consists of API-based approaches, based on components or widgets of the GUI. The advantage of using some kind of API provided by the OS or GUI library is that the recorded or modeled interactions are mapped to components or widgets of the GUI, instead of mouse coordinates, giving the second generation approaches better robustness against GUI changes [15]. In model extraction, the API-based approaches are more accurate than third generation visual approaches. Abbot [17] is an example of API-based C&R tool for Java GUI applications.

The third generation [15], or the third level of tolerance, approaches are based on VGT, using image recognition on partial images of the GUI and screen captures to interact and assert the correctness of the GUI. In some cases, VGT might be more tolerant to layout changes, but it is more dependent on the graphical representation of the GUI than API-based approaches. If the graphical icon of a button is redesigned, the related test cases have to be updated. An example of visual C&R tool is Sikuli [18].

The horizontal axis presents the level of automation in regression testing through the GUI. C&R approaches present the lowest level of automation, as manual effort is required both in recording of the test cases and in maintaining the test cases by re-recording test cases related to the changed parts of the GUI. The keywords or action words–based approaches would belong also to this first group.

The second level of automation is MBT using manually created models. This categorization addresses specifically regression testing, as manually created models may introduce a lot more benefits into other types of testing, such as testing if the GUI software conforms to the requirements specifications. The manually created models can provide a lot more information on the expected behavior, enabling better possibilities for generating test cases with meaningful test oracles. However, in regression testing, the effort required for manually creating the models is significant, and updating the models after changes in the GUI also requires some manual effort. Although TEMA tool [7] is not designed specifically for regression testing, it would fall into this category, and it is based on Android APIs to interact with the GUI.

The third level of automation is generating test cases based on automatically extracted models. In optimal cases, the level of automation with these approaches can be high. As described in Section II, the question is if the quality and efficiency of this testing is sufficient when the GUI changes, if the correctness of the model or the GUI is not assured with other means. API-based GUI model extraction and using the extracted models for test case generation has been a major topic in GUI model extraction and testing research during the last 15 years and there are a lot of academic tools available, such as GUITAR [19], GUI Driver [11], Testar [10], and Webmate [20], although Webmate has been commercialized. There is also more recent research and VGT GUITAR tool that is using visual approach in model extraction and test case generation [15].

The highest level of automation currently available is automated regression analysis using model comparison between automatically extracted models of the GUI software. With this approach, the manual effort remains in deciding if the reported GUI changes were intentional or regression faults. The only tool currently available in this category is open source Murphy tool [21].

IV. RELATED WORK

Since the evolution of automated GUI-based testing, presented in Section II, is a sort of state-of-the-art study, in this section we present related state-of-the-art studies in addition to related work on classifying automated GUI testing.

Kull [22] summarized the state-of-the-art on automated extraction of GUI models for the purpose of generating tests from the extracted models. The author raised the problem of not having meaningful test oracles as the main challenge in using extracted models for test automation.

Banerjee et al. [23] used systematic mapping to study 136 articles related to GUI testing to classify the nature of the articles, the aspects of GUI testing being investigated, the nature of evaluation being used, and to draw some conclusions based on the results. The authors conclude that more comparison is required between academic and industrial tools and techniques, and that commercial tools for MBGT are missing.

Aho et al. [24] presented an extensive state-of-the-art study on automated extraction of GUI models for testing. In addition to giving an extensive background on GUI testing and model extraction, the study summarized the work of most of the leading researchers and research groups related to using extracted GUI models for automated testing.

Alegroth et al. [15] have proposed to classify the existing GUI based testing approaches into three chronological generations. The first generation consists of C&R approaches capturing exact mouse coordinates. The obvious disadvantage, in addition to the general disadvantages of the C&R approaches, is the dependence on the screen resolution. If the same GUI software is executed on a different platform with a different screen resolution, the recorded test cases do not necessarily work.

The second generation consists of approaches based on components or widgets of the GUI and cover MBT approaches in addition to C&R. The advantage of using some kind of API provided by the OS or GUI library is that the recorded or modeled interactions are mapped to components or widgets of the GUI, instead of mouse coordinates, giving the second generation approaches better robustness against GUI changes [15].

The third generation approaches are based on VGT, using image recognition on partial images of the GUI and screen captures to interact and assert the correctness of the GUI [15]. There are also C&R tools, such as Sikuli [18], that fall into this category. In some cases, VGT might be more tolerant to layout changes, but it is more dependent on the graphical representation of the GUI. If the graphical icon of a button is redesigned, the related test cases have to be updated.

This mainly chronological classification [15] is not sufficient, as it does not address the level of automation at all, and all three generations have also C&R approaches. The most common inducement for adopting test automation is reducing the manual effort and time required for testing. Therefore the level of automation or amount of manual effort has to be considered when evaluation test automation methods and tools. Hence, in Section III we have proposed an improved classification of methods and tools for automated GUI testing having a second axis for the level of automation.

V. Conclusions, Discussion and Future Work

In this paper we summarized the evolution of GUI-based test automation and proposed a classification of methods and tools for automated regression testing through the GUI.

The classification proposed in this paper does not take into account all aspects of test automation related to testing through the GUI. Instead, it focuses on regression testing. Our intention is to provide a baseline for comparison between different tools and methods, and we hope that in the future the classification helps the industry in selecting the tools and methods most suitable for their needs. The variety of available test automation tools is growing, and it will become more challenging to select the tools that are best suited for the needs of a specific project.

Based on the state-of-the-art study, in the future we plan to address the lack of performance of GUI model extraction by executing the GUI being modeled in several virtual machines in parallel. Hence, we hope to get the automated regression analysis to be fast enough for the expectations of continuous integration processes. The same functionality could be used more generally to make automated UI test execution faster. We plan to work on combining component or API-based approach with visual image recognition aspects to make UI model extraction more accurate and tolerant for changes in the UI. Another future research subject would be using static analysis on the source code of the UI application to extract possible input combinations for increasing the coverage of model extraction.

References

[1] M. Grechanik, Q. Xie, and C. Fu, “Creating GUI Testing Tools Using Accessibility Technologies,” in International Conference on Software Testing Verification and Validation Workshops (ICSTW’09), 1-4 April 2009, Denver, CO, USA, pp. 243-250, IEEE.

[2] IEEE Standard Glossary of Software Engineering Terminology, 1990, IEEE, New York, USA.

[3] SeleniumHQ, a tool for automating Web testing, http://www.seleniumhq.org/ [retrieved: Apr, 2016]

[4] Appium, a tool for automating mobile app testing, http://appium.io/ [retrieved: Apr, 2016]

[5] Squish GUI Tester, http://www.froglogic.com/squish/gui-testing/ [retrieved: Apr, 2016]

[6] A. Kervinen, M. Maunumaa, T. Pääkkönen, and M. Katara, ”Model-Based Testing Through a GUI,” in Proceedings of the 5th international conference on Formal Approaches to Software Testing (FATES'05), July 11, 2005, Edinburgh, UK, pp. 16-31.

[7] TEMA Toolset, an MBT tool for mobile GUI testing, http://tema.cs.tut.fi/ [retrieved: Apr, 2016]

[8] B. Nguyen, B. Robbins, I. Banerjee, and A. Memon, “GUITAR: an innovative tool for automated testing of GUI-driven software,” in Automated Software Engineering, March 2014, Volume 21, Issue 1, pp 65-105, Springer.

[9] P. Aho, M. Suarez, T. Kanstrén, and A. Memon, “Murphy Tools: Utilizing Extracted GUI Models for Industrial Software Testing,” in Proceedings of 2014 IEEE International Conference on Software Testing, Verification, and Validation Workshops, 1-4 April 2014, Cleveland, OH, USA, pp. 343-348.

[10] T. Vos, P. Kruse, N. Condori-Fernández, S. Bauersfeld, and J. Wegener, “TESTAR: Tool Support for Test Automation at the User Interface Level,” in International Journal of Information System Modeling and Design, IJISMD 2015, vol.6 (3), July-September 2015, pp. 46-83.

[11] P. Aho, N. Menz, T. Räty, and I. Schieferdecker, "Automated Java GUI Modeling for Model-Based Testing Purposes,” in Proc. 8th Int. Conf. on Information Technology: New Generations (ITNG), 11-13 Apr 2011, Las Vegas, USA, pp. 268-273.

[12] Y. Miao and X. Yang, "An FSM based GUI test automation model," in Proc. 2010 11th Int. Conf. on Control, Automation, Robotics & Vision (ICARCV), Singapore, 7-10 Dec 2010, pp. 120-126.

[13] Windows Automation API, an interface for Windows GUI automation, https://msdn.microsoft.com/en-us/library/windows/desktop/ff486375(v=vs.85).aspx [retrieved: Apr, 2016]

[14] Jemmy framework, GUI library for Java GUI automation, https://jemmy.java.net/ [retrieved: Apr, 2016]

[15] E. Alégroth, Z. Gao, R. Oliveira, and A. Memon, “Conceptualization and Evaluation of Component-based Testing Unified with Visual GUI Testing: an Empirical Study,” in 2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST), 13-17 April 2015, Graz, Austria, pp. 1-10, IEEE.

[16] E. Elsaka and A. Memon, “A Fully Automated Approach for Debugging GUI Applications,” in 1st International Workshop on User Interface Test Automation (INTUITEST 2015), 19 October 2015, Sophia Antipolis, France, pp. 51-60.

[17] Abbot Java GUI Test Framework, http://abbot.sourceforge.net/ [retrieved: Apr, 2016]

[18] Sikuli Script, a visual GUI automation tool, http://www.sikuli.org/ [retrieved: Apr, 2016]

[19] GUITAR, a GUI Testing Framework, https://sourceforge.net/projects/guitar/ [retrieved: Apr, 2016]

[20] Webmate, fully automated web testing tool, https://webmate.io/ [retrieved: Apr, 2016]

[21] Murphy, an open source GUI model extraction, change analysis and testing tool, https://github.com/F-Secure/murphy [retrieved: Apr, 2016]

[22] A. Kull, “Automatic GUI Model Generation: State of the Art,” 2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops, 27 Nov - 30 Nov 2012, Dallas, TX, USA, pp. 207-212.

[23] I. Banerjee, B. Nguyen, V. Garousi, and A. Memon, “Graphical user interface (GUI) testing: Systematic mapping and repository,” in

Information and Software Technology, Volume 55, Issue 10, October 2013, pp. 1679-1694.

[24] P. Aho, T. Kanstrén, T, Räty, and J. Röning, “Automated Extraction of GUI Models for Testing,” in Advances in Computers, Vol. 95, 2014, pp. 49-112, Burlington: Academic Press.