COLLA 2018

The Eighth International Conference on Advanced Collaborative Networks, Systems and Applications

ISBN: 978-1-61208-645-3

June 24 - 28, 2018

Venice, Italy

COLLA 2018 Editors

Noriko Uosaki, Osaka University, Japan

Leonidas Anthopoulos, University of Applied Science of Thessaly (TEI), Greece

COLLA 2018

Forward

The Eighth International Conference on Advanced Collaborative Networks, Systems and Applications (COLLA 2018), held between June 24, 2018 and June 28, 2018 in Venice, Italy, aimed to gather an interdisciplinary spectrum of researchers around collaboration technologies.

Collaborative systems have raised to become an inherent part of our lives, supported by global infrastructures, technological advancements and growing needs for coordination and cooperation. While organizations and individuals relied on collaboration for decades, the advent of new technologies (e.g. from wikis to real-time collaboration, groupware to social computing, service-oriented architecture to distributed collaboration) for inter- and intra-organization collaboration enabled an environment for advanced collaboration.

As a consequence, new developments are expected from current networking and interacting technologies (protocols, interfaces, services, tools) to support the design and deployment of scalable collaborative environments. Current trends include innovations in distributed collaboration, collaborative robots, autonomous systems, online communities or real-time collaboration protocols.

The conference had the following tracks:

- Cooperation and collaboration mechanisms
- Machine learning applications for smart collaborative networks
- Smart and Connected Cities: Creating People-centered City Government

We take here the opportunity to warmly thank all the members of the COLLA 2018 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated their time and effort to contribute to COLLA 2018. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also gratefully thank the members of the COLLA 2018 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that COLLA 2018 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of collaborative technologies. We also hope that Venice, Italy provided a pleasant environment during the conference and everyone saved some time to enjoy the unique charm of the city.

COLLA 2018 Chairs

COLLA Steering Committee

Lasse Berntzen, University College of Southeast, Norway
Thomas Schuster, University of Applied Sciences Pforzheim, Germany

Mudasser F. Wyne, National University, USA
Wolfgang Mauerer, Technical University of Applied Sciences Regensburg / Siemens AG Corporate Research, Germany
Hassan A. Karimi, University of Pittsburgh, USA
Atsuo Hazeyama, Tokyo Gakugei University, Japan
Yung-Ting Chuang, National Chung Cheng University, Taiwan

COLLA Industry/Research Advisory Committee

Jingjing Wang, Google, USA
Niyati Chhaya, Adobe Research, India
Giuliana Dettori, Istituto per le Tecnologie Didattiche, Genova, Italy

COLLA 2018

Committee

COLLA Steering Committee

Lasse Berntzen, University College of Southeast, Norway
Thomas Schuster, University of Applied Sciences Pforzheim, Germany
Mudasser F. Wyne, National University, USA
Wolfgang Mauerer, Technical University of Applied Sciences Regensburg / Siemens AG Corporate Research, Germany
Hassan A. Karimi, University of Pittsburgh, USA
Atsuo Hazeyama, Tokyo Gakugei University, Japan
Yung-Ting Chuang, National Chung Cheng University, Taiwan

COLLA Industry/Research Advisory Committee

Jingjing Wang, Google, USA
Niyati Chhaya, Adobe Research, India
Giuliana Dettori, Istituto per le Tecnologie Didattiche, Genova, Italy

COLLA 2018 Technical Progrm Committee

Imad Abugessaisa, RIKEN, Japan
David Airehrour, Auckland University of Technology, New Zealand
Jameela Al-Jaroodi, Robert Morris University, USA
Hamed Alhoori, Northern Illinois University, USA
Mehdi Ammi, LIMSI-CNRS - University of Paris-Sud, France
Leonidas Anthopoulos, University of Applied Science (TEI) of Thessaly, Greece
Alessio Antonini, Università degli Studi di Torino, Italy
Marcelo Armentano, ISISTAN Research Institute (CONICET/UNICEN), Argentina
Liz Bacon, University of Greenwich, UK
Samia Ben Rajeb, Université Libre de Bruxelles, Belgium
Lasse Berntzen, University College of Southeast, Norway
Niyati Chhaya, Adobe Research, India
Jin-Hee Cho, U.S. Army Research Laboratory (USARL), USA
Christina Christodoulakis, University of Toronto, Canada
Yung-Ting Chuang, National Chung Cheng University, Taiwan
Neil Croft, University of Pretoria, South Africa
Antonio De Nicola, ENEA, Italy
Giuliana Dettori, Istituto per le Tecnologie Didattiche, Genova, Italy
Arianna D'Ulizia, National Research Council - IRPPS, Italy
Hossein Fani, University of New Brunswick, Canada
Camille Fayollas, ICS-IRIT | University Toulouse 3 Paul Sabatier, France
Carlos Granell, Universitat Jaume I of Castellón, Spain

Copyright Information

Table of Contents

Supporting Collaborative Interaction among Learners Using Collaborative Learning System InCircle

Noriko Uosaki
Center for International Education and Exchange
Osaka University
Osaka, Japan
Email: n.uosaki@gmail.com

Takahiro Yonekawa
Brain Signal, Inc.
Tokyo, Japan
Email: yonekawa@bsgnl.com

Chengjiu Yin
Information Science and Technology Center
Kobe University
Kobe, Japan
Email: yin@lion.kobe-u.ac.jp

Abstract—**This paper describes the development and the evaluation of a collaborative communication system called InCircle. One notable trend in the pedagogical field is a shift from a teacher-centered learning to a student-centered one. Recent advancement of Information and Communication Technology (ICT) has accelerated the shift. The spread of various mobile applications for educational use has enabled us to share information and knowledge in real time. In order to promote student-centered learning, it is necessary to facilitate interaction among learners. For these purposes, InCircle was introduced in a university level language and culture class in Japan. The objectives of our research study were to examine whether InCircle contributes to the facilitation of interaction among learners and the enhancement of mutual cross-cultural understanding. The evaluation was conducted in an international exchange subject where international students and Japanese students learn about the language and culture at university in Japan. The result of the questionnaire showed the learners' preference to InCircle as opposed to the blog comment function. It also endorsed its usability and effectiveness as a communication tool.**

Keywords- collaborative learning; InCircle; international students; international exchange subject; student-centered learning.

I. INTRODUCTION

In recent years, one notable trend in the pedagogical field is that more teachers have moved towards a student-centered learning environment. In the course of this shift, student-centered collaborative learning has drawn much attention from researchers in the pedagogical field [1]. Collaborative learning has a "social constructivist" philosophical base, where learning is regarded as construction of knowledge within a social context [2]. It is reported that student-centered and small-scale course programs resulted in more academic success than lecture-based course program [3]. It is also reported that a student-centered collaborative learning is one of the most effective ways of learning in language classes [4]. In fact, most studies investigating the link between the extent to which course programmes are student-centered on the one hand and promote academic success on the other hand, find positive relationships between the two [2].

Recent prevalence of high-performance mobile devices has enhanced the potential of learners' active interaction via mobile-based communication tools. We have seen a good deal of researches on communications applications, such as educational application of social network service (SNS) [5]-[8]. Our communications application project, InCircle, is among them. In this study, InCircle was introduced for the purpose of facilitating interaction among learners and enhancing learning opportunities.

The rest of the paper consists of the following sections: Section 2 explains our developed system, Section 3 explains the experiment setup and Section 4 explains the conclusions and future works.

II. INCIRCLE

InCircle [9] is a product developed by AOS Mobile Inc., Tokyo, Japan [10] with the second author joining this project as a chief software architect. It is a client-server application. The server side runs on Linux OS and Windows Server. The client side is working on iOS, Android, and PC Web browser. Chat messages are transmitted and received through the network (Figure 1).

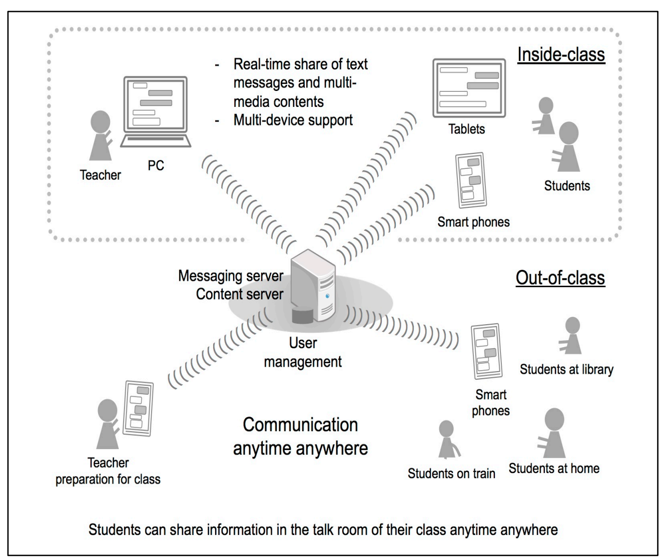


Figure 1. InCircle system configuration.

 ISBN: 978-1-61208-645-3

The system allows users to create groups. Group members are able to send and receive messages and multimedia files in their chat room with an easy operation. Chat messages are synchronized in real-time to realize smooth communication. Figure 2 shows a chat room interface when the instructor posted an interrogative sentence: "Do you have textbook authorization system in your country?" since interrogative sentences trigger active interaction among learners which leads to mutual cross-cultural understanding.

In our system, we have mainly four major advantages:

a) Teachers can be administrators of the system.

Teachers can be administrators of the system so that they can watch the users/their students' behaviors. Therefore, they can avoid their students' malicious behaviors via InCircle. In case of inappropriate behavior from the part of a student, teachers can delete or close the student's account.

b) Teachers can pre-register user accounts.

Unlike other SNS or chat tools, such as Facebook and LINE, user accounts are pre-registered. Teachers create accounts for their students and make a group for the class in advance. There are always some students who do not want to use the existing SNS systems. In fact, in one of the authors' classes, some students rejected to create a Facebook account and some students did not want to use LINE. Unless all the students agree to use it, it is not appropriate to use it as a communication tool in class. Besides, the existing SNS users usually post their private information on their profiles. However, some students may not wish to share private information with some of their classmates. In our system, on the contrary, there is no page in the first place to fill in their private information, such as school career, birth place/date.

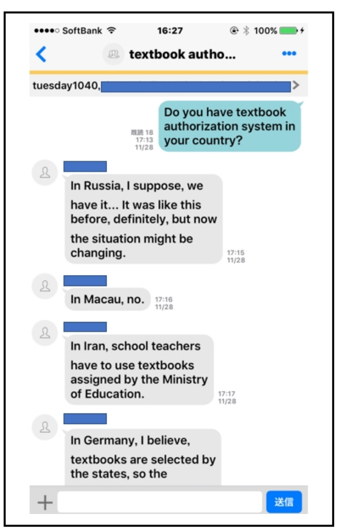


Figure 2. InCircle chat room interface.

c) Security is ensured

Every effort was made in order to ensure the security, such as encryption of the cache data in the client terminal, channel coding, encryption of database, the use of different cryptography keys for each company or school in the server side. Therefore, it is highly protected against divulging of information or account hacking.

d) Users can delete the sent messages.

In our system, we can delete messages after they are sent not only on the sender's side but on the recipient's side. It is likely to happen that we send messages by mistake. Our system can handle such human errors.

Our research questions are:
(1) Can InCircle contribute to the enhancement of the students' mutual cross-cultural understanding?
(2) Can InCircle contribute to the enhancement of interaction among the students?

III. EVALUATION

A. Target class

The target class was one of "international exchange subjects" which was targeted mainly for international exchange students. Japanese students who are interested in class held in English can also join it. The class was held 12 times on a once-a-week-basis in a CALL (computer assisted language learning) room during the fall/winter semester, 2017. The class language was mainly in English. The objectives of the target class were (1) to enhance cross-cultural understanding and (2) to improve the skills of their target languages, which were Japanese or English. The class consisted of 21 students (6 Japanese, 4 Germans, 2 Chinese, 2 Finnish, 1 British, 1 French, 1 Korean, 1 Macao, 1 Russian, 1 Taiwanese, 1 Vietnamese). All the participants were owners of mobile phones. At the beginning of the semester, they were asked to answer the learning style questionnaire created by Academic Skills Advice service [11] to identify their preferred learning styles.

B. Procedures

InCircle was introduced in class from November 14, 2017 to January 23, 2018. It was introduced mainly because not all the students had the identical SNS accounts, such as Facebook, WhatsApp, LINE, Twitter and so on.

A class blog was created by the teacher as a communication tool. Google Blogger service was used for creating the class blog. It was used throughout the semester from October 3, 2017 to January 23, 2018. The teacher posted contents which were useful for classroom learning as well as each class schedule. The mailing list was also created by the teacher for the facilitation of class communication. However, even though the teacher instructed the students to use a class mailing list as a communication tool among classmates, it turned out that it did not play any role as a communication tool.

 ISBN: 978-1-61208-645-3

Therefore, the mailing list was used only by the teacher when she needed to share necessary information with the students.

Towards the end of the semester, when students used to both communication tools, i.e., InCircle and Blogger, close examination was made on how the students used them as communication tools during the class held on January 9th, 2018. For the enhancement of the mutual cultural understanding, the teacher posted theme topics on both sites. Since it was a class held right after the new year holidays and for most of the students, it was their first new year holidays in Japan, the topic was new year holidays: "Did you find something special for お正月(New Year)?" on Blogger site and "How do you celebrate new year holidays in your country?" on InCircle. The students were instructed to post comments on both media. They were informed that their posting would affect their grades.

C. The Results

Table 1 shows the number of posts in InCircle talk room and the blog comment columns. The number of posts on InCircle was 26, while that of Blogger was 14. The number of the participants on January 9th' class was 20. The number of the students who posted comments on InCircle was 18, while those on Blogger was 8. As the number of the post shows, the students were more active in posting messages on InCircle.

TABLE I. COMPARISON BETWEEN INCIRCLE AND BLOGGER COMMENTS ON JAN. 9, 2018 (PARTICIPANTS: 20)

InCircle	Blogger
How do you celebrate new year holidays in your country?	***Did you find something special for お正月(New Year)***
The number of posts	The number of comments
26 (18 students)	14 (8 students)

Tables II and III show the participants' comments reacting to the teacher's questions via both media. Figure 3 shows the interfaces of both media. As for Blogger comment function, its users cannot post multimedia files so that the information they could get was only text-based.

One notable phenomenon for InCircle was an interaction among students, which did not happen in Blogger's comment column. One comment (No.15) posted by Student #13 triggered Student #4's question (No.18), which was answered by Student #13 (No.21), which triggered Student #17's photo post (No.22). This interaction ended with Student #4's post showing her gratitude (No.24) as shown in Table III. Apparently, InCircle promoted the interaction among students. On the contrary, when the teacher posted a question in Blogger site, Did you find something special for お正月 (oshogatsu: New Year), 8 students posted comments in the comment column. This topic was expected to stimulate the students' interactions since for the most students, the new year 2018 was their first new experience in Japan, and in fact, new year holidays are the most special holidays in Japan. There must have been many things new to them. But unlike the teacher's expectation, one student's post did not lead to another students' reaction. They just answered to the teacher's question (Table II).

TABLE II. BLOGGER USE ON JAN. 9, 2018

No.	Time 1/19/2018		Comments
	11:12	Teacher	**(Q) Did you find something special for お正月(New Year)**
No.1	16:37	Student #1	**(A)**おもち(omochi: rice cake)
No.2	16:37	Student #2	**(A)** しめ飾り(shimekazari: new year decoration made from straw ropes
No.3	16:38	Student #3	**(A)** I played with kites on Oshogatsu when I was a kid
No.4	16:40	Student #2	**(A)**お雑煮(ozouni: New Year soup with rice cake)
No.5	16:40	Student #4	**(A)**年賀状(nengajou* New Year Card)
No.6	16:40	Student #5	**(A)**It's very interesting that there is a particular importance of doing something for the first time in the beginning of the new year (like watching the first sunrise or first visiting temple).
No.7	16:41	Student #6	**(A)** お年玉 (Otoshidama: New Year money to give children)
No.8	16:41	Student #7	**(A)**おせち料理. (osechiryouri: New year food
No.9	16:42	Student #2	**(A)**初詣(hatsumoude: New Year's visit to a Shinto shrine)
No.10	16:43	Student #2	(A) 初日の出 (hatsuhinode: First sunrise)、書初め(kakizome:New Year's writing)
No.11	16:43	Student #7	(A)kakizome
No.12	16:45	Student #8	(A)初売り(hatsuur:The first sale)
No.13	16:55	Teacher	(Related to No.12) 福袋（ふくぶくろ :fukubukuro:Lucky bags ） https://en.wikipedia.org/wiki/Fukubukuro
No.14	16:55	Student #2	(A)かるた(karuta:cards)

(Q):question (A):answer

TABLE III. INCIRCLE POSTS ON JAN. 9, 2018

No.	Time 1/19 /2018		Comments
	10:27	Teacher	**(Q) How do you celebrate new year holidays in your country?**
No.1	16:26	Student #1	**(A)** In Germany we celebrate with fireworks on New Year Eve and the first day of the new year. We usually congratulate everyone on midnight, when the new year starts.
No.2	16:26	Student #2	**(A)** We eat Berliner (German sweet bread filled with jam or alcohol) and do fireworks at midnight
No.3	16:35	Student #3	**(A)** A lot of my friends and family in Germany do a fondue at New Year Eve because it takes quite long to eat and that night you have the time for a long dinner.
No.4	16:44	Student #4	**(A)** We celebrate the Spring Equinox (March 21st, 春分の日) as the first day of the New Year.
No.5	16:46	Student #5	**(A)** In the night of 31st of December, we eat 年越しそば (toshi-koshi Soba)
No.6	16:46	Student #4	photo related to No.4

No.7	16:47	Student #6	**(A)** In the UK people spend New Year eve drinking with friends and family. We count down at midnight, sing "auld lang syne" (traditional Scottish song) together and then watch the national fireworks display.
No.8	16:48	Student #7	**(A)** in Finland we have fireworks around midnight and we usually spen New Year with friends. Sparkling wine is common and https://www.asejaera.fi/image_view.php?name=U/listaus_Tahtisadetikku-19-cm-sadetikku-1.jpg
No.9	16:48	Student #6	**(A)** In Italy we eat lentils on New Year eve as it is supposed to make you rich over the next year, we also drink and celebrate with family
No.10	16:48	Student #8	**(A)** In China, we rarely celebrate for New year and we only have 3 day off. However, we will celebrate for Chinese New year, celebrating with family.
No.11	16:48	Student #9	**(A)** In Japan we watch 紅白歌合戦 on TV at 31st of December.
No.12	16:48	Student #10	**(A)** In France we usually party with friends and/or family, drink champagne, do the count down and there is also fireworks.
No.13	16:48	Student #11	**(A)** I'm Japanese but since my family has lived in many places, we've just settled for not doing much in New Year.
No.14	16:50	Student #12	**(A)** For us (in Russia) fireworks are so important during the New year night, that they are usually associated with this fest particularly. While preparing the food for the dinner we watch some "traditional" Russian movies about the New year. After the midnight we exchange presents (by searching them underneath the Christmas tree) and go out to walk around. It's very easy to meet all friends on the streets at night.
No.15	16:50	Student #13	**(A)** In Macau, we will go count down, but it is not a main festival for us. The main one, Chinese New Year, sits in February, and we will have reunion lunch or dinner and receiving red pockets.
No.16	16:50	Teacher	**(Related to No.11)**紅白歌合戦（こうはくうたがっせん） https://en.wikipedia.org/wiki/K%C5%8Dhaku_Uta_Gassen
No.17	16:52	Student #14	**(A)** In Taiwan we eat dinner with family and play fireworks in 12/31.And we go to temple in 1/1. But we celebrate the lunar calendar.
No.18	16:52	Student #4	**(Q to No.15)** @Studner#13: what are those red pockets?
No.19	16:52	Teacher	**(Related to No.5)** 年越しそば（としこしそば） https://en.wikipedia.org/wiki/Toshikoshi_soba
No.20	16:52	Student #15	**(A)** During Chinese New Year, family gather together and have dinner, watch TV, after 12 o'clock of new year's eve, older generation will give out red pockets. 1/9/2018 16:52
No.21	16:52	Student #16	**(A to No.20)** Red pockets are like お年玉 in Japan.
No.22	16:52	Student #17	**(Related to No.21)** Red pocket's photo
No.23	16:53	Student #18	**(A)** Buy fireworks and use them in 31.12 after klo 18.00. Burn candles, ULKOTULI
No.24	16:54	Student #4	**(A to No. 21)** I see! Thanks.
No.25	16:54	Student #18	ULKOTULI's photo **(related to No.23)**
No.26	16:56	Teacher	**(Related to No.12)**お年玉（おとしだま）

(Q):question (A):answer

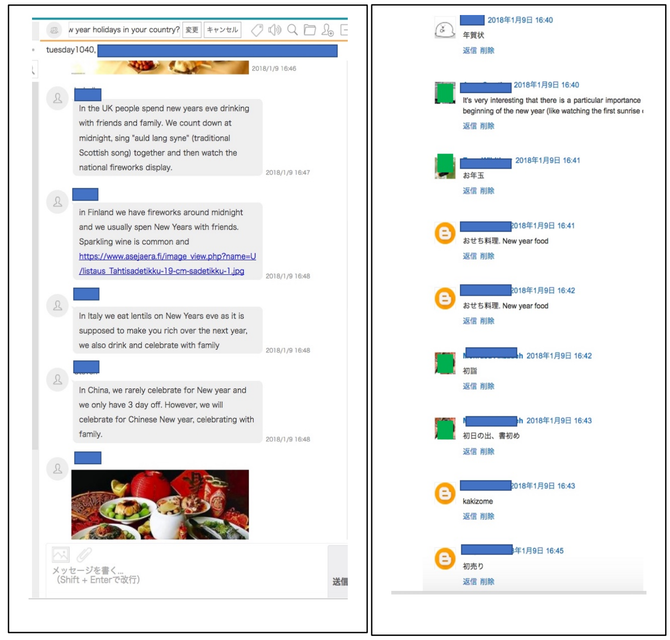


Figure 3. InCircle posts (left) and Blogger comments (right)

At the end of the phase, they were asked to answer the five-point-scale-questionnaire as shown in Table IV. Q1 and Q3 were created based on the technology acceptance model proposed by [12]. Q2 was created to examine the fun factor of our system. Q4 and Q5 were created to examine its contribution to the class objectives. Q6 and Q7 were created for examining the user acceptance of its interface and the whole system.

TABLE IV. THE RESULTS OF THE 5-POINT-SCALE QUESTIONNAIRE

	Questions	Mean	SD
Q.1	Was it easy for you to use InCircle?	4.4	0.86
Q.2	Was it fun for you to use the system?	3.9	0.89
Q.3	Was it helpful as a means of communication with your classmates and teacher?	3.7	1.19
Q.4	Was it helpful for understanding Japanese culture and other cultures?	4.3	0.90
Q.5	Was it helpful for your target language learning?	3.2	0.94
Q.6	Please rate its interface	3.8	0.60
Q.7	Please rate the whole system.	3.7	0.80

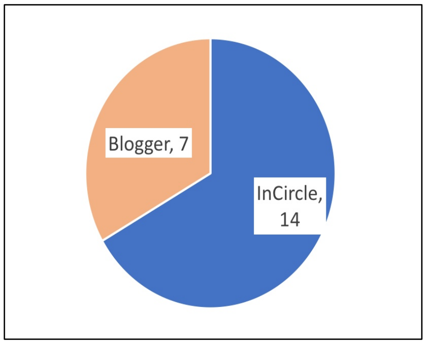


Figure 4. Which was easier to handle?

The highest score, 4.4 was given when they were asked about the usability of the system (Q.1). As some students felt it was like other existing apps (as in comments #2, 9 and 13 in Table 5), most students had already used other similar apps, which lead to the high score of the usability of InCircle. The lowest score, 3.2 was given when they were asked if it was helpful for their target language learning (Q.5). Since it was a basic rule in 'International Exchange Subjects' to run the class in English, the language in InCircle was mostly English. Therefore, it was helpful for English language learners (6 students), but it was not so helpful for Japanese language learners (15 students). Since the improvement of their target language skills is one of the class objectives, it is our next work to consider how our system can be utilized for the language learning including its contents. The average scores of its fun factor (Q2), its interface (Q6) and the whole system (Q7) were 3.9, 3.8 and 3.7. which we consider them passable enough so that no urgent refinement is necessary.

Figure 4 shows the result of the question: which one was the easier to handle, InCircle or Blogger site? two thirds (14 out of 21) of the students felt InCircle was easier to handle. This implies that the majority felt that the Blogger site was more complicated to write comments in than InCircle. It was in line with the high score of Table 4 Q1.

D. Discussion

Our research questions were (1) Can InCircle contribute to the enhancement of the students' mutual cross-cultural understanding? (2) Can InCircle contribute to the enhancement of interaction among the students? As for (1), it can be said safely that our system contributed to enhancement of mutual cross-cultural understanding because of the fact that the average point of Q.4 (Was it helpful for understanding Japanese culture and other cultures?) was as high as 4.3. In traditional teacher-centered lecture style class, it is difficult for the students to learn from other students. For instance, they could learn how to celebrate New Year in other countries by using InCircle. As for (ii), even though the average point of Q.3 (Was it helpful as a means of communication with your classmates and teacher?) was 3.7, not as high as expected, more students used InCircle than Blogger comment column and the number of postings via InCircle outnumbered Blogger comments (Table I). Therefore, it can safely be said that our system contributed to enhancement of interaction among the students. Since other SNS tools, such as Facebook, LINE, twitter have some weaknesses as a communication tool for classroom use as mentioned earlier, InCircle is expected to be an effective communication tool in student-centered collaborative learning class.

Table 5 shows the participants' free comments on InCircle.

TABLE V. THE STUDENTS' IMPRESSION OF INCIRCLE

No.	Comments
#1	so so
#2	it's good
#3	Quite similar to any chat system I experienced before...
#4	It's a good system and easy to use.
#5	It seems like a neat tool with a clean interface and without much bloat.
#6	Good for communication
#7	I like how easy it is to access.
#8	i would use it as part of class
#9	Pretty good but similar to other platforms
#10	Just normal chatting Website
#11	It could get a little disorganized with so many people in the class.
#12	Good Experience
#13	I think it is like the application, "LINE" or "Skype".
#14	It was refreshing and fun! New style!
#15	I'm not sure how exactly I would benefit from using the site. It is nice and all, but we did not use it that much in class so that I would have gotten a clear understanding in what kind of situations it would be helpful. Connecting with teacher is for me done by email or in person, and classmates you would use whatever social media app is popular atm, email or in person.
#16	There is, in my opinion, no need to use it. It provides you with an easy opportunity to communicate with class members and the teacher but there is no need for communication. This class is about computers in language learning, and in my opinion sharing with others is not the primary way of achieving that. Therefore, I am just not interested in using these kind of programs.
#17	Very useful!

There are two clearly negative comments: #15 and #16 which seem non-negligible. The result of the learning style questionnaire which was conducted at the beginning of the semester, showed their preference of "learning alone" to "learning in a group". There might be a possibility that their negative attitude toward group learning reflected their negative comments on InCircle. This is not directly connected with the effectiveness of the system itself, but how to deal with those students who prefer "learning alone" is an important issue to cope with for an implementation of a successful collaborative learning.

IV. CONLUSIONS AND FUTURE WORKS

In this study, we describe facilitating interaction among students using InCircle. When compared with Blogger comment column, InCircle showed its superiority in many

 ISBN: 978-1-61208-645-3

aspects as described in Discussion section. The questionnaire results showed that the students were satisfied with its usability. Our hypotheses (research questions (1) and (2)) were proved to be correct. However, it was found out that our system was not supportive for their target language learning. Since it is one of the two objectives of the target class, it is necessary to consider how to introduce InCircle from the pedagogical viewpoint in order to support their language learning. It is among our future works to find out some solutions to improve the skills of their target languages via InCircle. In addition, even though it does not concern our system itself, there are some students who are not keen on collaborative learning. How to get those students involved in collaborative learning is another issue to tackle in the future.

ACKNOWLEDGMENT

Part of this research work was supported by the Grant-in-Aid for Scientific Research No.16H03078 from the Ministry of Education, Culture, Sports, Science and Technology (MEXT) in Japan.

REFERENCES

[1] Y. Li, "Student-centered Approach and Communicative Language Learning in the Exam-oriented EFL Settings," In Proceedings of the 3rd International Conference on Foreign Language Learning and Teaching 2013 (FLLT 2013), pp. 471-482, 2013.

[2] R.L. Oxford, "Cooperative Learning, Collaborative Learning, and Interaction: Three Communicative Strands in the Language Classroom," The Modern Language Journal 81, 1 pp. 443-456. DOI: 10.1111/j.1540-4781.1997.tb05510.x

[3] S. Severiens, M. Meeuwisse, and M. Born, "Student experience and academic success: comparing a student-centered and a lecture-based course programme," Higher Education: the International Journal of Higher Education and Educational Planning 70(1), pp. 1-17, 2015. doi:10.1007/s10734-014-9820-3

[4] T. X. Chen, "Interactive Learning of a Foreign Language," Journal of Acoustic Society of America, 114(1), 30, 2003. DOI: 10.1121/1.1601085

[5] C. Munoz and T. Towner, "Opening Facebook: How to Use Facebook in the College Classroom," In I. Gibson, R. Weber, K. McFerrin, R. Carlsen and D. Willis (Eds.), Proceedings of Society for Information Technology and Teacher Education International Conference 2009, pp. 2623-2627, 2009.

[6] T.-H. Hung and S. C.-Y. Yuen, "Educational use of social networking technology in higher education," Teaching in Higher Education 15(6), pp. 703-714, 2010.

[7] S.-Y. Kim and M.-R. Kim, "Educational Implication of Reflection Activities Using SNS in Cooperative Learning," In Proceedings of the 13th International Educational Technology Conference, pp. 340-347, 2013.

[8] K.T. Kwak, S. K. Choi, and B. G. Lee, "SNS flow, SNS self-disclosure and post hoc interpersonal relations change: Focused on Korean Facebook user," Computers in Human Behavior, 31 pp. 294-304, 2014.

[9] InCircle: https://www.incircle.jp/ [retrieved: June 2018]

[10] AOS Mobile Inc., Tokyo, Japan: http://www.aosmobile.com/ [retrieved: June 2018]

[11] Academic Skills Advice service: www.brad.ac.uk/academic-skills / [retrieved: June 2018]

[12] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," MIS Quarterly, 13 (3), pp. 319–339, 1989.

 ISBN: 978-1-61208-645-3

The Integration of Home Collected Data into the Veterans Administration Health System

The Home Telehealth VistA Integration Project: The Validation of New Components

George C. Blankenship, Jr.
Consultant to Home Telehealth IT PMO
Alexandria, Virginia
e-mail: BlankenG@GWU.edu

Ellen Hans
U. S. Department of Veterans Affairs
Home Telehealth IT PMO
Silver Spring, Maryland
e-mail: Ellen.Hans@VA.gov

***Abstract*—In 2004, the US Department of Veterans Affairs (VA) was confronted with the creation of an enterprise application based upon multiple triage systems supplied by vendors who had dramatically different levels of experience with system integration or with the Health Level 7 International Standard (HL7). One objective of the Home Telehealth application was to leverage the capabilities of the existing VA Electronic Medical Record (EMR) system and medical software wherever possible. To accomplish this objective, a strong integration approach was required. The selected approach was the development of a reference engine that could be used to validate the communication between existing VA medical applications and the newly procured triage systems. This paper describes the validation approach used to evaluate a procured system's readiness to operate in the collaborative environment of VA's enterprise medical record system of over 200 distinct independent systems. The paper represents the opinion of the authors and is not a statement of any official position of the VA.**

Keywords-telehealth; home telehealth; telemental health; HL7; CCOW; protocol validation; realtime HL7 transactions; collaborative processing; system validation.

I. INTRODUCTION

The introduction of new components into an existing mature distributed environment is a complex problem. The objective is that the new component performs its task reliably and without adverse impact on the distributed environment. To achieve this result there must be a thorough system validation prior to placing the system in production. It is generally acknowledged that key to a well-behaving system is adherence to a System Development Life Cycle (SDLC) model; that the creation of a well-formed requirements document set, a well-formed testing plan, and the development of a testing laboratory. The requirements document must be complete and accurate; however, all documentation is stale the minute it is published. The documentation can be validated by a test the existing systems; this validation is beyond the subject of this paper. The validation plan for the new components should cover all the transactions with cases that include normal expected valid messages and exceptional conditions, such as those that create retransmissions and those that indicate an error. The testing must include the building and processing of messages, the movement of message, and the building and processing of transactions; that is, the testing sequences must cover both normal sequences (error free) and exceptional sequences (error indications). With the testing of both valid processing sequences and the handling of exceptional conditions, there is a confidence that the existing environment will not be impacted by the introduction of a new or updated component. This paper addresses the validation testing of an addition of a new telehealth component to an existing mature distributed medical distributed system. The validation of the new system's compliance with security requirements is a separate subject.

Section II of this paper introduces the major systems involved in the home care and the interaction between the systems. Section III describes the process used to validate that a procured system is ready to support patient care. Section IV discusses the acceptance testing challenges for the procured system. Section V lists the major documents used to support the integration. Section VI introduces the primary validation tool set. Section VII discusses the lessons learned and future tasks.

II. HOME BASED CARE

A. Background

The United States Department of Veterans Affairs (VA) [1] Home Telehealth program is clinically managed by the Office of Telehealth Services (OTS) [2]. The program is supported by the VA's Office of Information & Technology (OIT) [3] Home Telehealth Program Management Office. The program uses a telemedicine approach for the care of patients with a chronic illness such as diabetes, congestive heart failure or chronic obstructive pulmonary disease. Care is also provided for patients suffering from post-traumatic stress disorder. A patient's current medical condition is collected using Interactive Voice Response (IVR) scripts, a specialized home device, and/or an intelligent mobile device. The collected information is sent to a triage system in a VA data center. The triage system analyzes the submissions from each patient and creates a prioritized list for the care nurses; that is, the triage system orders the nurse's medical care. The nurse uses the clinical desktop (Graphical User Interface/GUI) to monitor the patient condition and direct patient care for a panel of over one hundred patients.

B. VA Health Care

The Veterans Health Administration (VHA) is America's largest integrated health care system, providing care at 1,243 health care facilities, including 170 medical centers and 1,063 outpatient sites of care of varying complexity serving 9 million enrolled Veterans each year. VHA has an integrated health system based upon a distributed Electronic Medical Record (EMR) system that allows clinical personnel to access all patient data that has been collected at any VHA medical facility in the US and abroad. Each of the medical centers hosts an instance of the medical record system. The core component of the health system is the Veterans Health Information Systems and Technology Architecture (VistA) [4]; the clinical desktop for a VistA system is Computerized Patient Record System (CPRS). A patient's data can be presented as an aggregation independent of the original visit location. The VHA's approach to patient care is based upon the ability to access a patient's complete medical record from any VHA location.

C. The Home Telehealth Triage System

The core component for the care of a patient in the home is a triage system supplied as a Commercial off the Shelf (COTS) system with the required modifications to meet the interface to VHA systems. The COTS portion addresses the telemedicine componentry:

1. the devices used by the patient to collect their current medical state,
2. the system that performs the triage function to rank the patient needs, and
3. the transmission components that transfer the collected data to the triage system.

The development of the interface to the existing VHA systems is a task for the supplier; it would be very uncommon for a supplier's system to support HL7.

The VA supplies the necessary documentation defining the HL7 encoding specifics and all the transactions used to allow the integration of the supplied triage system with the VistA environment. All communication with VA systems is based upon HL7 messaging as defined in the HL7 2.4 standard with the specific definition of each HL7 message field as would be required for any usage of a standard.

The distributed environment is an autonomous environment where systems perform an automatic recovery from exceptional conditions. This expectation is a normal requirement for systems have an associated urgency, such as a medical care system. The movement of each message includes confirmation of receipt within an appropriate period; if the receipt confirmation is not received, the message is retransmitted automatically. A transaction is normally moved in a single message; however, some transactions require multiple messages. The processing of a transaction generates a completion status message; that is, the transaction was, or was not, processed successfully. Once again, if the completion message is not received within an appropriate period, the transaction will be retransmitted. The consequence of these processing rules is that the overall system can operate autonomously with no manual intervention; manual intervention has a tendency to introduce undesired delay and unexpected problems. A consequence of autonomous operation is that a violation may create havoc. A misdirected message may cause indefinite retransmissions or an output queue stall; for example, a misdirected receipt confirmation would confuse the receiving system while causing the proper system to not empty its output queue causing a stall. A system that does not retransmit a message due to a missing receipt confirmation will also experience an output queue stall.

D. VistA Integration

The telehealth component of the program is based upon a strong binding between the triage system and the devices used by the patient; that is, the triage system provider supplies the devices used by the patient and the analysis engine (the triage system itself). VHA procures new triage systems periodically so that the latest technology is available. With each new award the program is confronted with the integration of multiple triage systems developed by suppliers with a varied amount of integration experience or HL7 experience. Each triage system supplier also wishes to keep abreast of technology and clinical desires by providing new capabilities. Depending upon the complexity of an upgrade, a new system validation may be required for such technology upgrades. The clinical team may require a new set of data be sent from a triage system to the VHA EMR; that is, the clinical team may require a change in the VHA EMR environment interface. The upgrade would also require a new validation of the triage system along with an update of the documentation.

The objective of the triage system integration with the VHA EMR is to leverage the capabilities of the existing VHA EMR and medical software wherever possible. The care providers (doctors and other clinicians) should not be required to access the clinical desktop of the triage system.

The objective of the Home Telehealth VistA Integration project is the integration of the Home Telehealth triage systems into the VHA distributed processing environment. This is a real time sharing of data between an individual triage system and the VHA EMR systems. This integration approach starts with the establishment of a patient record in the triage system using VHA identifiers and the synchronization of the patient identity using the VHA identity management system. With a consistent identity, the triage system can send patient data collected at the home to VHA systems in a computable form so that the data can be used in standard VHA patient care analysis packages. The triage system presents the collected information to VHA systems such that the information has the look and feel of data collected by any other VHA medical facility. The data is then available to all VHA medical personnel and all the VHA medical outcomes analysis processing engines without requiring access to the triage system itself. The end result of the integration requirement is the home care component of the patient's medical program is not an isolated data island, but is a system integrated into the VHA's approach to patient care.

III. VALIDATION STRATEGY

The fundamental elements of the SDLC approach is based upon

1. the development of a document [7] that defines all the transactions used by the Home Telehealth program,
2. a reference engine that implements all the transactions and processing rules defined by the document, and
3. a validation test plan.

The document supplements the HL7 and CCOW standards with the specific VHA encoding definitions and processing rules.

The reference engine emulates a triage system to existing VHA systems to validate the transactions and encoding definitions. The reference engine also emulates VHA systems to a triage system to validate the conformance and operation of the triage system's HL7 engine and transaction processing. This use of a single reference engine presents a reliable base for the evaluation of the triage system integration into the mature VHA distributed processing environment.

The triage system validation is performed in three phases. The first two phases are directed by a set of functional validation objectives in two different testing laboratories. The third phase is a slow start deployment of patient care to minimize risk. The focus of this document is the first phase of the validation.

The first phase is performed in an External Integration Test Laboratory (EITL). The EITL contains a full emulated environment of the VHA systems and software that would peer with a triage system. The EITL is accessible via the Internet allowing the System Under Test (SUT) to be in its own laboratory. The experience gained since the program’s inception in 2005 is that successful real time processing (autonomous processing) of transactions requires reliable, predictable and consistent performance of the HL7 engine and transaction processor in the triage system. These required characteristics are achieved by the proper handling of exceptional conditions that occur during transmission and other events that occur during normal system operation. The objective of the EITL laboratory testing is to verify that an SUT is capable of the real time processing requirements required for VistA Integration in the presence of anomalous conditions.

The second phase is performed in the VHA Integration Test Laboratory (ITL). The ITL contains a full environment of the VHA systems and software that would peer with a triage system in production. In the ITL, an SUT is evaluated while supporting test patients. VHA clinicians evaluate the SUT clinical desktop; VHA IT staff evaluates system logs for proper system performance. Anomalous conditions are not generated in the ITL unless there is a defective application.

The third phase is a limited national release (controlled release) of a production version using a slow start approach. The production system is first brought into use at a single VHA facility for Home Telehealth patient care for a five-week evaluation period. A second and the third facility are added to the environment at weeks three and four as the testing progresses. The operation is evaluated at the end of each week. At the end of evaluation period, national use is authorized for all 170+ VHA major medical facilities and their subsidiary treatment centers

IV. Acceptance Testing Challenges

A. *Program Start Up Consequences*

The program started in 2004 as a set of pilot programs ended and the national program started. The program approach was based upon the procurement of a set of triage systems from suppliers experienced in providing remote home health care, but little experience with HL7; thus, the initial stand up did not include VistA Integration. Patient identity and demographics information was manually entered into the triage systems. The impact of the manual entry of the patient identity was keystroke errors and the copying of the wrong information from other system displays. Unreliable data entry creates a problem when a clinician wishes to correlate data in the triage system with data in the VHA EMR. Our research found that the error rate for manually entered patient information was 7%. The end result of the manual entry of patient identity in the triage system was the duplication of patient records and orphaned patient records in the triage system. The impact of the errors has haunted the program since its inception; the identification of orphan records is a difficult task.

B. *Acceptance Testing Approach*

It is well recognized that acceptance testing is required for a distributed processing environment. This testing is generally a duplication of the Software Quality Assurance (SQA) testing of software in the second phase of the SDLC model. VHA has a testing lab that mimics the operational (production) environment. It includes

1. multiple medical record systems instances (VistA),
2. an identity management system (Master Patient Index/MPI or Master Veteran Index/MVI),
3. a medical database system (Health Data Repository/HDR),
4. a patient census and survey repository system (Census and Survey/CNS),
5. a patient database, and
6. other various systems.

The focus of this lab (Integration Test Lab/ITL) is the SQA testing of VHA applications and the validation testing of procured systems. There is no capability to generate abnormal conditions since the ITL is a replication of the operational environment (“real world”). No system in the operational world intentionally generates exceptional conditions; no system creates conditions that require a recovery from an exceptional condition. It is this property that prevents an environment, such as the ITL the deep inspection for the validation of a new system and some system upgrades. Due to these limitations, the Home Telehealth program decided to create the EITL which is designed to generate anomalous conditions during a test.

 ISBN: 978-1-61208-645-3

C. VistA Integration Challenges

VistA Integration uses messages defined by Health Messaging Level 7 International Standard [5] (HL7) to communicate with VHA medical systems and HL7 Clinical Context Object Workgroup [6] (CCOW) defined application context control to integrate the triage system clinical desktop with the standard VHA desktop applications. VHA has an extremely mature HL7 environment. The first challenge of the integration is the verification that a triage system has an HL7 engine implementation that would allow the integration. The emulator was developed to perform the verification. The second challenge is the implementation of a CCOW environment for the triage system clinical desktop. The solution of these two challenges allowed the program to start the piloting of VistA Integration.

D. Data Collection Challenges

The triage system receives data from multiple sources within the home. The use of this data requires that the triage system maintains an appropriate data quality inspection identifying a questionable submission. The triage computation is based on an accurate collection from the home. The data sent by the triage system to the VA databases must be an accurate representation of the patient state. The triage systems are time synchronized with the VHA systems and all end devices are synchronized with time servers either directly or via the triage system. However, some of the devices can be used in a disconnected mode with internal power and a manual configuration. The user might have to initialize the end device. The disconnected operation of end devices allows for the device to lose time synchronization with the triage system. The device itself might fail and present an incorrect reading. The patient might make a transcription error when the reading the value for a manual entry. Entries with an error are sent to VA databases with an indication of “entered in error”; thus, the entry will be tracked, but not used.

E. Autonomous Processing Challenges

The system concept is that the triage systems operate autonomously. The guiding tenet for any system must be based upon Jon Postel’s guidance “an implementation must be conservative in its sending behavior, and liberal in its receiving behavior.” Every system must focus on the generation of correct and accurate messages; however, a system must be prepared for the messages received from the real world. The real world is imperfect and capable of corrupting transmissions in an unpredictable manner.

F. Validation Testing Approach

The Home Telehealth program is providing daily care to patients that are not in a medical facility. The objective of the program is to improve the medical condition of the patient or provide the necessary medical care that will address the patient’s current condition. It is most important that the readings are presented in a timely manner along with the triage analysis is available promptly. The system must be able to deal with an exceptional condition, such as a communication error, a message with questionable data, or processing that is delayed. All the exceptional conditions mentioned above have led the program to adopt a more exhaustive approach to the acceptance criteria of a triage system.

V. Documentation

The guiding document for the integration of a procured triage system with the VHA EMR is the *Home Telehealth HL7 Functions Overview* [7]. The *Home Telehealth VistA Integration Test Plan* defines the individual tests. All documentation is available upon request.

VI. Reference Engine

The development of the reference engine started in late 2004 when research indicated that there was no publicly available software package that would validate the processing of the transactions used to integrate an external system with the VHA processing environment. The testing tool must provide for the ability to tailor each HL7 message field for a thorough inspection of the message processing.

VistA Integration forces the triage system to slave patient identity to the MPI and not allow changes from any other source including the triage system clinical desktop. There is the requirement to validate that the SUT will properly process an update of each component of an HL7 identity. The difficulty of the construction of the required tests to perform this task has the natural consequence of ignoring the test. The reference engine was developed to perform such a task.

TABLE 1. HOME TELEHEALTH TRANSACTIONS

HL7	Source	Peer	Function
ADT-A04	VistA	Triage	Create a new patient record and/or activate patient care
ADT-A03	VistA	Triage	Inactivate patient care
QBP-Q22	Triage	MPI	Register with the MPI
ADT-A24 **ADT-A31** **ADT-A43**	MPI	Triage	Patient record update
MDM-T02	Triage	VistA	Progress Note
MDM-T02	Triage	Survey	Patient Survey Response
MDM-T02	Triage	Census	Triage System Census
ORU-R01	Triage	HDR	Patient Vitals Submission

The reference engine (commonly call the emulator) includes the ability to perform as either a single triage system (triage system emulator) or a set of systems that exist the distributed medical system (VistA, MPI, HDR, Census and Survey systems emulator). This approach guarantees that the reference engine is a system using the same operational rules for the communication with every system that exists in the distributed processing environment while allowing it to perform inspections of diverse components of the environment.

The transaction set used in the Home Telehealth program are listed in Table 1. The **HL7** column lists the HL7 message used for the transaction. The **Function** column describes the purpose of the transaction. The **Source** column indicates which system of the distributed processing environment would generate the transaction. A triage system emulator would be used to validate the processing rules described in the program documentation or participate in the testing of a new application or capability; Section VI.A describes the operation and objectives of a triage system emulator. A medical system emulator would be used to validate the processing by a triage system or generation of transactions by the triage system; Section VI.B describes the processing and objectives of a medical system emulator.

A. Triage System Emulation

The triage system emulator was used to validate that the *Home Telehealth HL7 Functions Overview* did indeed reflect the operation rules for the VHA systems. The development was started in late 2004 and the initial validation of the program documentation was completed in early 2005. The capability of the reference engine to generate any transaction with all appropriate HL7 message movement proved to be extremely useful and supplied a unique tool for development teams. This capability was first used in the spring of 2005 to aid the development of the VistA component that receives patient progress notes from external systems.

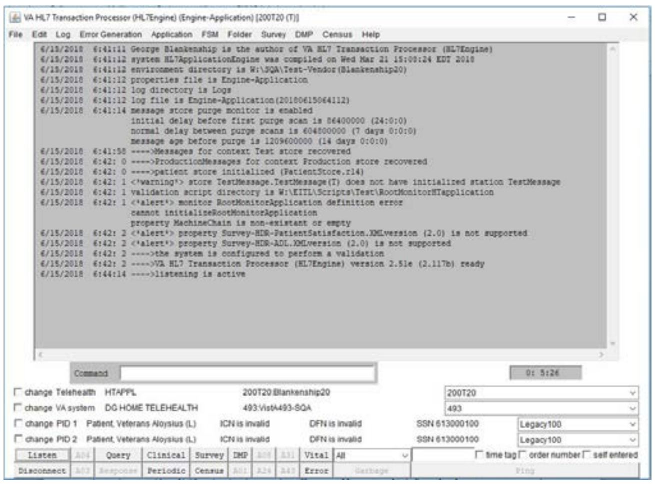


Figure 1: Reference Engine GUI (Triage System Application)

Figure 1 shows the GUI (desktop) for the triage system emulator. The figure shows the current GUI as of 2018; there have been many revisions since the emulator was used in 2005. At the bottom are six rows of widgets used to select configuration items and transactions to be generated. The triage system facility number is shown in the first row. The second row identifies the peer system to receive a transaction generated by the emulator. The third and fourth rows identify the patient or patients that are in a transaction. The bottom two rows are used to select an asynchronous transaction to be sent to the peer system (see Table 1). The GUI is common for both emulation modes and buttons that are not applicable in the current mode are disabled. A button from the bottom two lines is used to select the HL7 message to be sent. Some exception conditions can be injected by use of the **Error Generation** menu item at the top of the GUI.

The triage system emulator processes all transactions autonomously as would a real triage system. The following paragraph describes the processing of a patient activation by a triage system.

The ADT-A04 transaction establishes a new patient to be serviced or activates an existing patient for service. The transaction is sent by a VistA system to a triage system. The triage system must register with the MPI. A successful registration would allow the activation to be successfully completed by the triage system. The processing rules include the following three items.

1. The processing of the transaction must not cause a data conflict in the triage system database.
2. The MPI can identify the patient; that is, the registration is successful.
3. All components of the patient identity are consistent with the MPI.

This transaction is used as an example due to its complexity and the global impact of a violation. If the rules are not followed, multiple peer applications will not be able to process transactions from the triage system. There are three unique identifiers used in the VHA medical record: a VA national identifier, a US government national identifier, and a medical center identifier. Every medical record must contain the same set of identifiers for patient identification. The Home Telehealth program uses the VA national identifier for patient identification. It sends transactions to other applications that use one of the other two identifiers. With the successful processing of the activation message, the triage emulator will be able to process received transactions.

B. Medical Record System Emulation

The original objective that started the development of the reference engine was the necessity to validate that a procured triage system did, indeed, meet the requirements. From the beginning it was obvious that the only approach was a "black box" approach; that is, it would not be possible to instrument the SUT. The validation engine started out in 2005 with the generation of correct data in all transactions sent to the SUT and the ability to force retransmissions.

The 2005, a validation required that each test was controlled using the widgets. The validation started in May of 2005. The National Rollout of the first validated triage system started in August of 2005. During the first three years of the program rollout, conditions were encountered that led to the consideration the development of an automated test generation and scoring component; time and resources always postponed the work. Most of the conditions are discussed in the earlier sections of this paper. For a system to be reliably capable of error recovery, a test tool must exist that can create the exceptional conditions on command. Every few years new triage systems were procured. With each procurement the test coverage was expanded. With the improvements and correction were made over the years, the validation was strengthened. There was one glaring problem; the testing was a manual effort. Each test was built through the GUI. The scoring of the test generally required a manual examination of the system log file. The process was very tedious. At the same time, while the improvements did uncover latent problems, the most vexing problems remained as troublesome issues in production. The problems were of a very low occurrence in the production world and did not apparently cause patient care issues; thus, the problems remained a discussion items in weekly conferences and buried by larger issues. The problems seemed to indicate that the autonomous processing was flawed in selected systems. A possible explanation over the years was improper recovery or improper processing sequences.

For a system to be reliably autonomous, a test tool must exist that can generate the conditions that force a system to choose a transaction sequence order; the sequence should comply with the Atomicity, Consistency, Isolation, and Durability (ACID) database principles. The use of more extensive testing required a major enhancement to the emulator. In 2014, a major enhancement to the validation component of engine was finely started. The enhancement was the addition of scripting component that would allow more complex tests to be generated with a minimal effort and the automatic scoring of tests results. Before the development of the scripting component, the validation was performed using 140 distinct tests. Each test had distinct sequence of manual operations performed by the test conductor on the emulator GUI. Due to time constraints, a successful validation required two to three months; if any issues were uncovered, the validation period increased. In late summer of 2014, the first usable component of the scripting was completed and used for an ongoing validation. The addition was based upon the use of a publish and subscribe component in the core of the emulator. This component was always intended to instrument a system based upon the core software but had not been used in the emulator. The addition of a component that subscribed to change in the processing state of the engine allowed for the monitoring of messaging flow; the scripts supply the expected state using a finite state machine approach and expected conditions. The result was the ability to generate a transaction sequence with scoring.

The enhanced emulator immediately detected that the current SUT had ACID violations. The SUT had been in service since 2005 and had just undergone a major revision requiring a re-validation. Over the years, it experienced some strange problems that defied resolution or reproduction. The system was finally identified a fatally flawed. The development of the scripting was completed in early 2015. The impact of using scripting was dramatic. The scripts for the almost 1,000 distinct tests were developed in a month. The tests are bundled into test sets. The test conductor starts a test set. The test set identifies the preconditions for the testing. After the preconditions are established, the tests run autonomously. As each test completes, an error conditions generates an email to the triage system administrator. Four new triage systems were validated in 2017 in a 6-month period.

VII. Conclusion

System validation in a distributed environment is a difficult and tedious task. System components are developed in isolation. The tools to build software components do not have the capability to reproduce the distributed environment; the testing focus is generally on the user interface (GUI) with a limited capability to simulate the communication with a remote system. Many test cases just cannot be covered; many test cases cannot be defined.

The development of the Home Telehealth emulator enabled the validation phase of the program SDLC to approach a realistic evaluation of the readiness of a system's capability to support patient care in the VA medical network. The major enhancement of the testing scripting in 2014/15 allowed the actual validation of proper processing in the presence of exceptional conditions. Unfortunately, the avoidance of the developing the necessary software is realistic; the Home Telehealth emulator is over 1,000,000 lines of Java code (over 450 Java classes).

The most dramatic impact on the validation was the release of the validation tool to the triage suppliers. Until the suppliers installed an instance of the testing tool in their lab, their testing was impacted by scheduling. Each supplier had to have a dedicated testing slot with publicly available validation engine in the EITL.

The observations derived from each validation of updated or new components are not unique to the medical world. All distributed processing environments would benefit from a concerted effort to develop or acquire the necessary tools to perform a comprehensive evaluation of additions before the addition is joined to the environment.

A new validation round is currently scheduled for the 2019/20 period with the acquisition of a new set of triage systems providing the latest technology. During the intervening period the validation test set will be enhanced to cover more problems that have vexed the program over the years. VHA is also planning to add the use of the HL7 standard Fast Healthcare Interoperability Resources (FHIR) to the EMR. This effort will start a new major update to the Home Telehealth emulator. The initial research for the emulator's support for FHIR should start in the summer of 2018.

References

[1] United States Department of Veterans Affairs. "U.S. Department of Veterans Affairs". Available from: http://www.va.gov/ [retrieved: 4,2018].

[2] United States Department of Veterans Affairs. "VA Telehealth Services". Available from: http://www.telehealth.va.gov/ [retrieved: 4, 2018].

[3] United States Department of Veterans Affairs. "VA|Information Technology". Available from: http://www.oit.va.gov/ [retrieved: 4, 2018].

[4] Department of Veterans Affairs Office of Enterprise Development, "VistA-Health*e*Vet Monograph", 2008; Available by request from the authors.

[5] Health Level Seven International. http://www.hl7.org/2; Available from: http://www.hl7.org/ [retrieved: 4, 2018].

[6] Clinical Context Object Workgroup. Available from: http://www.hl7.org/Special/committees/visual/index.cfm [retrieved: 4, 2018].

[7] George C. Blankenship, "Home Telehealth HL7 Functions Overview", U.S.D.o.V. Affairs, Editor 2018; Available by email to George@GBlankenship.us.

Cross-Platform Development

Suitability of Current Mobile Application Frameworks

Jan Christoph, Daniel Rösch, Thomas Schuster
Hochschule Pforzheim
Tiefenbronner Straße 65, Germany
{jan.christoph | daniel.roesch | thomas.schuster}@hs-pforzheim.de

***Abstract* — Mobile device adoption has increased dramatically within the last decade. In addition to smartphones, wearables and various sensors are among the most utilized devices. At the same time, the multiplicity of devices increases, the number of platforms to consider when developing applications increases as well. It is therefore desirable to be able to generically develop applications and deploy them to all relevant target platforms. A typical approach is given by frameworks, which generate platform specific code. In this article, we examine the suitability of these frameworks. Central questions are access to native system functions, sensors of devices and support for upcoming platform developments. To evaluate the frameworks, we defined a reference application and implemented tests for different mobile devices and platforms. A final framework comparison reveals opportunities and limitations. This, in turn, serves as a foundation for future work on improvements of promising approaches.**

Keywords-cross-platform; app development; Web engineering; component-based software architectures.

I. INTRODUCTION

Mobile devices have become an important platform for today's software applications. Especially, the utilization of smartphones increased rapidly within the last couple of years [1], [2]. Since smartphones are often utilized to consume or orchestrate services, this process includes a vast range of applications; they also connect to other domains such as the Internet of Things (IoT) and utilize smart cloud-based services.

The introduction of smartphones rapidly increased the need and development of mobile software. The development of mobile software applications is a special case of software engineering. Mobile applications are often also referred to as apps, which implies that the application is intended to be used on a smartphone or wearable device [3]. Thus, development must cope with specific aspects such as: short application lifecycles, limited device capabilities, mobility of users and devices, availability of network infrastructure as well as security and privacy issues [4]. While developers are enacted to create and distribute applications in a large scale, they also have to deal with these inherent limitations of mobile devices (i.e. battery life or small displays). Furthermore, it is necessary to address different operating systems (especially for smartphones, and, to a limited extent, for feature phones as well). Since the market for smartphones has consolidated recently, some operating systems (i.e. Windows Phone, BlackberryOS and other OS hold a market share of 0.2%) vanished again. Still, to address the smartphone market, applications for both, Android (market share: 85%) and iOS (market share: 14.7%) need to be provided. In addition, Android is split into different versions, manufacturers and various system customizations. Currently, the most widely used Android version is Nougat (7.0 and 7.1 with 28.5%), while little use is made of the newest Version Oreo (8.0 and 8.1; with 1.1%) [5].

To reach as many users as possible, all major platforms and versions need to be supported [4], [6]. This introduces the need to either develop platform specific or platform agnostic applications. Platform specific implementations (native apps) require as many application implementations as platforms are intended to be addressed. Therefore, this approach generates correspondingly high development expenditures without additional added value. On the other hand, with a more generic approach, a single application or some core components could serve as the basis for multiple platforms. Besides reduced developments efforts, a generic approach also strengthens reuse of code and components.

Currently, generic approaches can be further subdivided into Web and hybrid applications (see Fig. 1). Web applications can be used virtually under any platform, as a Web browser is preinstalled on almost all devices. The most salient advantage is application portability, which basically comes at no cost. Web apps are typically optimized by means of Hyper Text Markup Language (HTML5), Cascading Style Sheet (CSS) and JavaScript [7]. Numerous frameworks (such as Angular, Bootstrap, React or Vue) provide additional functionality on top of Web standard technologies and help to speed up development of Web apps. Major disadvantages of Web applications are that they do not possess platform specific look and feel and often are restricted in functionality – especially access to system functions and device sensors. Furthermore, they must be interpreted and suffer performance losses compared to native applications [8].

Hybrid applications are built on frameworks such as Apache Cordova or Adobe PhoneGap. Often they rely on Web technologies also, and enact access to native device functions and sensors [4]. Hybrid apps utilize a specialized browser to present the user interface (UI). This results in a presentation layer which is identical or very near to widgets used in native apps. While hybrid apps overcome some

 ISBN: 978-1-61208-645-3

issues of Web apps (such as access to system functions and sensors), they still experience a loss of performance compared to native applications. However, it is notable that performance of hybrid apps has improved a lot with latest developments [4], [7]. Comparing the short development lifecycles of devices and operating systems on the one hand to that of hybrid app frameworks on the other, it is noticeable that the latest developments are implemented with delays by the frameworks. As a result, access to new functionalities can be gained earlier when development is based on native apps.

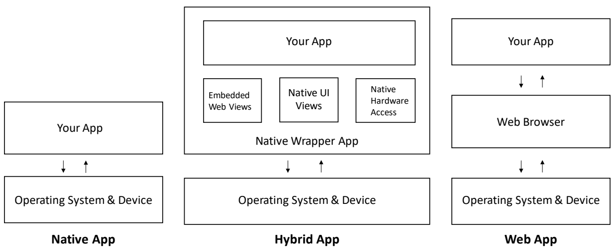


Figure 1. Mobile App Technology Stack

Issues of supported functionality, performance and the generic question of maintenance of cross-platform applications lead us to the evaluation of multiple cross-platform frameworks. The remainder of this article is structured as follows: Section 2 provides an overview of current mobile app development. In Section 3, a reference architecture is presented and three framework-based implementations of this architecture are discussed. The reference implementations are being evaluated in Section 4. Finally, Section 5 provides our conclusion and outlook on future work.

II. RELATED WORK

A. Cross-Platform Development

As stated above, there are several approaches for cross-platform development. This type of development is subject to typical challenges of ubiquitous computing. In addition, further challenges are typical to cross-platform development [4], [9], the most important being associated with:

1. UI
2. Limited Resources
3. Device Management
4. Application Maintenance

The design of UI is associated with questions of simplicity and intuitiveness. For mobile cross-platform development, this is extended by design guidelines defined by the different operating systems. It is further restricted because of different device capabilities (e.g., screen sizes and resolution) [10]. Limited resources is a typical issue in mobile software engineering; for cross-platform development the application size and resource consumption (especially power and memory management) is a typical issue [4], [11]. Since cross-platform development addresses a vast variety of devices, their management in terms of appropriate usage of hardware and sensors (i.e. cpu, memory, bluetooth, or camera) becomes another typical challenge. Furthermore, different operating systems must be handled as well. Finally the application has to be maintained by following short lifecycles of devices, operating systems and frameworks [4], [10].

A lot of different methods that address cross-platform development can be observed in science and industry. Some are based on model-driven software engineering [12]. The advantage of model-driven methods is, that developers and users which are less familiar with specific programming paradigms are enabled to efficiently implement applications. As Object Management Group (OMG) standard, the Interaction Flow Modeling Language (IFML) offers model-based and platform-independent development of applications for different types of devices. Following the Model-Driven Architecture (MDA) it is based on a meta-model and it is built upon Web Modeling Language (WebML). A Web-based and an eclipse-based modeling environment is provided for IFML. Furthermore, extensions for Apache Cordova and PhoneGap are provided [12]. An open challenge is to keep the extensions up-to-date. Other solutions, such as WebView, utilize native code and combine it with Web technologies. Native components are used as containers to render Web pages that contain application logic and presentation layer definitions. Native components serve to access device-specific functions (i.e. push notifications or sensor data). Although WebView is a native application, it can internally use Web technologies without switching to a standard browser. WebView also supports CSS and JavaScript for custom interface development [8]. However, WebView does have two main drawbacks: 1) custom styling is necessary to gain a native look and 2) its performance is below average [13]. In summary, we observe three general approaches to cross-platform development:

1. Native Application
2. Transformation- or generator-based Application
3. Interpreted Application (Parser-based)

With native development, an application is developed for each specific device (and operating system). Benefits include the native look and feel, the ability to use all platform-specific features and a comparatively high performance of the app. The most prevalent disadvantage is high efforts for development and maintenance. The latter is a result of redundancy in code and support because each platform has to be served by a separate application [8], [9].

The use of generators employs a meta-implementation which is then transformed to specific platforms (e.g. as used in Cordova or Ionic). Similarly, model-driven development approaches (such as IFML) may use transformations to produce platform specific code. An advantage is that the application logic is platform agnostic [12]. Applications which are interpreted rely on some kind of parser. The parser interprets application code during runtime in order to create platform specific instructions. Fabrik19 utilizes an interpreted approach in its Mobility Suite (MOS) framework.

 ISBN: 978-1-61208-645-3

B. *Cross-Platform Frameworks*

As discussed above, there are a lot of cross-platform frameworks like IFML, Cordova, Corona Software Development Kit (SDK), Appcelerator Titanium, TheAppBuilder, PhoneGap, Native Script, SenchaTouch, Framework7, Apache Weex, Flutter, Jasonette or Manifold – also see [6]. All of them utilize one or a combination of the three methods to create platform specific applications. In our comparison, we strive to evaluate the most frequently used and most progressively developed frameworks (see Fig. 2).

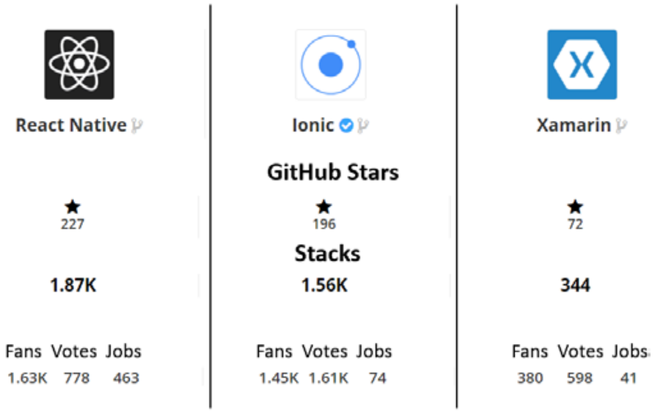


Figure 2. Ionic vs. React vs. Xamarin [14]

Ionic offers a generator-based approach [15]. The framework is free to use and available as open source. Additionally, several services are available via pay on demand. The generator utilizes a Web application as input. Thus, development of cross-platform applications is based on Web technologies (JavaScript/TypeScript, HTML5 and CSS; see Fig. 3). Ionic also relies on Angular [15] in order to foster component based development and reuse of templates. Ionic officially supports Android, iOS and UWP [16]. Since Ionic is based on Web applications that are generated into platform specific applications through Apache Cordova, these source applications may also be executed in any Web browser. Native operating system functions and access to sensors is only available after generation of platform specific code. The utilization device specific functionalities often also rely on plugins that have to be declared as dependency [17].

Figure 3. Ionic Architecture

Xamarin is another framework to develop cross-platform apps for Android, iOS and Universal Windows Platform (UWP) [18]. Other platforms such as Linux are not supported and MacOS support was recently added with the launch of Xamarin.Mac.

Xamarin is based on .Net and utilizes C# as programming language. Xamarin is divided into two major parts: 1) Xamarin platform and 2) Xamarin.Forms. The Xamarin platform (Xamarin.Android, Xamarin.iOS) provides APIs to share code for application logic between all platforms. The UI is written individually for each platform. Xamarin.Forms allows to create additional platform-independent UI, which are mapped into native UI in a second step. The development environment is based on Visual Studio (or Xamarin Studio for macOS) [18].

React Native is a parser based open-source framework for building cross-platform applications [19]. It is based on React. Both frameworks are being developed by Facebook. React Native currently supports Android and iOS. However, with a little more effort, it is also possible to deploy to UWP. Since React is built on JavaScript, this holds true for React Native as well. React Native invokes Objective-C APIs to render to iOS components and Java Application Programming Interface (APIs) to render to Android components. This means that no code generation is utilized in React Native. Facebook promises that the performance of apps would be almost as good as that of native applications. Components for React Native may either be built as functional components or class components [19].

III. REFERENCE ARCHITECTURE AND IMPLEMENTATION

This paper follows the constructivist paradigm of design science [20]. Thus, insights will be retrieved by creating and evaluating artifacts in the form of models, reference architectures and, in our case, specific implementation variants and efforts spent on their creation. Contrary to empirical research, the goal is not necessarily to evaluate the validity of research results with respect to their truth, but to the usefulness and feasibility of the different approaches in order to solve a common problem – here, to deploy with ease to different mobile platforms. Following this line of thought, requirements will be imposed by the definition of a reference application architecture. The reference architecture is derived using common hypotheses, practitioner interviews and literature review. The reference architecture serves as requirements model for the implementation of different alternatives and tests in a real environment.

Thus, the reference application architecture is defined to compare most utilized frameworks against each other and to identify strengths and weaknesses. To enact a comprehensive comparison [6], [21], the application should access native system functionalities and provide a platform specific UI. In short, the frameworks should generate applications, which are close to native applications. Thus, we also evaluated against platform specific UI guidelines for Android and iOS [22]. We defined the following functional reference criteria:

1. Layout: Grid
2. Layout: Tab
3. Operating System Function: Access current time
4. Sensor Function: Access current position (GPS)

 ISBN: 978-1-61208-645-3

5. Sensor Function: Access the phone camera

In addition to functional criteria, it is also important to measure quality aspects, such as development efforts and application performance. Therefore, we analyzed two different types of layouts mentioned in the list above, that are often used in today's apps – Mockups are depicted in Fig. 4.

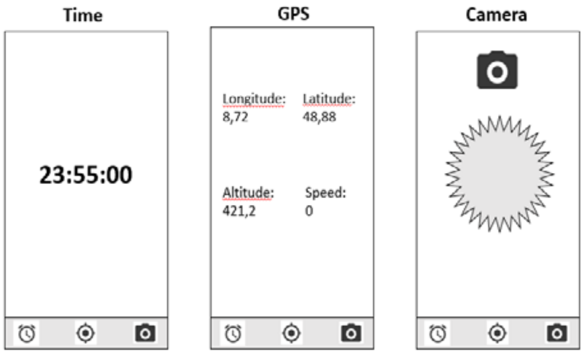


Figure 4. Wireframes

A. Ionic

Layout – Grid: Ionic provides a typical Grid-View with the `<ion-grid>` component [16]. Furthermore, styling of the GridView can be set individually. **Layout – Tab**: Using Tabs in Ionic is easy as wall, it may be just derived by use of the starter template (which provides this from scratch). Precise instructions may also be found in the documentation [16].

Access system time: This is derived by simple and built-in JavaScript function calls (e.g., date().getHours() is used to get the current hour). **Access current position (GPS)**: To determine the position, the Cordova plug-in Geolocation has to be installed via npm. Then, it can be integrated in the project [16]. As shown Listing 1, the position can be retrieved, if the necessary sensors are available and permissions are given. **Access to the camera**: To use the camera, the Cordova plugin Camera is required and has to be integrated into the project [16].

Listing 1

```
getThePosition(){
   this.geolocation.getCurrentPosition().
      then((resp) =>{

         this.longitude = resp.coords.longitude;
         this.latitude = resp.coords. latitude;
         this.altitude = resp.coords. altitude;
         this.speed = resp.coords. speed;
   }).catch((error) => {
      console.log("Error getting location", error);
   });
}
```

Debugging & testing: Ionic offers several methods to debug and test apps. If the application is not utilizing sensor information, a clean Web test can be driven (by `ionic serve`). Web tests may be carried out as known for Web applications in general – such as debugging by means of the browser's developer console (`F12` shortcut) or employing Web driver test scripts. If sensor information is utilized the application has to be deployed to a platform specific device or an emulator. With Ionic this can be done by calling `ionic cordova build android|ios` to build the app and `ionic cordova emulate android|ios` to execute the app on an emulator. If a test device is being utilized instead of emulation (by calling `ionic cordova run android|ios`) the application may again be tested in a browser, e.g. using Google Chrome (`chrome://inspect/#devices` has to be called and the specific device has to be selected). In order to automate unit testing typical tooling as known for other JavaScript-based frameworks can be used. To test the reference implementation, we could simply employ the well-known frameworks Karma and Jasmin.

B. Xamarin

Layout – Grid: In Xamarin the layout differs, depending on the chosen platform. For Android `GridView` and for iOS `uicollectionview` has to be used [18]. **Layout – Tab**: In Xamarin tabs have to be set up manually. There is no standard template available to support this layout. Typically, a tabbed page will be used to reference other content integrated as tabs.

Access system time: To retrieve the system time, a `ViewModel` is created, and a `DateTime` attribute tracks the current time. For updates a `PropertyChanged` event is fired. The reference is made possible by the data binding. **Access current position (GPS)**: The current position is determined by the plugin `Xam.Plugin.Geolocator` [18] (installed via NuGet). Adjustments are needed to support Android. In addition, necessary privileges for querying the position must be granted. After configuration, the logic can be implemented. Attributes for longitude and latitude have to be mapped to determine the location (see Listing 2). **Access to the camera**: Camera access is realized with the plugin Xam.Plugin.Media [18]. It has to configured by means of xml. In important step is the definition of a resources folder to determine where to store captured pictures and videos. The camera itself can be called asynchronously (`getTakePhotoAsyncCommand`).

Listing 2

```
public async System.Threading.Tasks.Task
   getLocationAsync()
   {
     var locator = CrossGeolocator.Current;
     locator.DesiredAccuracy = 50;
      if (locator.IsGeolocationAvailable &&
            locator.IsGeolocationEnabled) {
              var position = await
                locator.GetPositionAsync();
              this.Longitude="Longitude" +
                position.Longitude.ToString();
              this.Latitude="Latitude" +
                position.Latitude.ToString();
       }
   }
```

Debugging & testing: Xamarin enables unit testing and debugging with Visual Studio. For Xamarin, Visual Studio basically offers the same mechanisms as known for any other

 ISBN: 978-1-61208-645-3

component which is developed within Visual Studio (such as break points and live debugging). Visual also offers support for asynchronous testing and mock object creation, e.g. if the Model View Viewmodel (MVVM) pattern is applied and view models invoke service operations asynchronously. Visual Studio also provides a well sophisticated profiler, which provides monitoring of memory utilization and object allocation. Finally, Xamarin also offers also a test cloud for UI-Tests – where automated testing for native and hybrid applications is done by employing the App-Center.

C. React Native

Layout – Grid: React Native does not provide a grid layout immediately. To resemble a grid-layout within the reference implementation, a ScrollView component was used and individual views had been adapted by means of CSS. Alternatively, third-party grid components could be utilized as well to resemble a grid layout. React Native Easy Grid and React Native Layout Grid are just two examples of these components, which may be installed via npm. **Layout – Tab**: React-Native Expo IDE can create a starter app which directly operates with tabs. Manual creation is not as easy as in Ionic but efforts are still considerably low.

Access system time: Is achieved by simple JavaScript calls. `this.state` [19] is needed for the databinding and `new Date().getHours()` retrieves the current hour. **Access current position (GPS)**: The determination of the current position is already integrated in the React Native API [19]. The position is retrieved by calling `navigator. geolocation.getCurrentPosition`, further details can be seen in Listing 3. **Access to the camera**: Camera and access rights have to be configured and `hasCameraPermission` has to be set to zero. The `componentWillMount` method the permissions are checked and we the status is updated. The asynchronous method `takePicture` is utilized to check if the camera is available and if it was possible to take a picture.

Listing 3

```
Navigator.geolocation.getCurrentPosition(
   (position) => {
      this.setState({
         latitude: position.coords.latitude,
         longitude: position.coords.longitude,
         error: null,
      });
},
(error) => this.setState({ error: error.message }),
{ enableHighAccuracy: true, timeout: 20000,
maximumAge: 1000},
);
```

Debugging & testing: React Native similarly offers multiple ways to debug and test apps. Debugging mode can be activated from a developer menu. This can be called by keyboard shortcuts or, if running on a test device, by shaking the smartphone. To debug the JavaScript code in Chrome, a remote debugging session can by created when select `Debug JS Remotely` is selected from the developer menu. This will open `http://localhost:8081/debugger-ui` in a new browser tab. Other debugger implementations may be used as well and a recommendation then would be to use the standalone version of React developer tools. These can be installed via `npm install -g react-devtools` and may be called via `react-devtools`. To set up unit testing for React Native it is recommended to utilize Jest and execute tests via node. For integration testing, several different options exist. Integration testing always relies on platform specific environments; thus those have to be set up first.

IV. EVALUTION

The evaluation and comparison of different frameworks is based on our test app. We selected evaluation criteria based on the evaluation framework developed by Heitkötter et al. [23]. It covers different evaluation criteria, especially for infrastructure (including the lifecycle as well as the functionality and usability of the app) and app development (including testing, debugging and developing the app). We also extended or removed some criteria (e.g. scalability) to focus the following app properties:

1. Supported platforms
2. Supported development environment
3. Access to platform-specific functions
4. Application look and feel
5. Application portability
6. Simplicity of development
7. Application performance

It is important which platforms (Android, iOS, UWP, etc.) and to which extent these are supported by each framework. The next criterion discusses all possible development platforms and environments (Windows, MacOS and Linux). With the help of our test app we intend to analyze if platform-specific functions are available. Also, an evaluation of the UI is conducted to measure platform specific look and feel. Moreover, we want to unveil if the source code is reusable and if it can be integrated into other frameworks (portability). Also, the development efforts play a major role and will be evaluated within criterion 6. In order to assess and evaluate efforts and feasibility of the frameworks, we asked five experiences developers to implement our test application according the reference architecture. The following evaluation is also based on their feedback. Finally, we conducted an assessment of the application's performance. Therefore, the test app is used to measure: start time, used memory and execution speed of internal functionalities such as GPS polling. For this purpose, three test devices (Honor 9, Sony XZ1, Samsung Galaxy S7) were used. 100 test runs were conducted for each device to stabilize results.

A. Ionic

Configuring a system for Ionic and creating a first app only takes a few minutes. Regarding ramp up, the majority of our developers found that Ionic is the easiest framework to start with. It has to be mentioned, that it is necessary to ensure that all dependencies (to plug-ins) are installed according their declared version. This can be error prone, especially when Ionic is updated. In case of multiple app

 ISBN: 978-1-61208-645-3

development projects, conflicts may also arise between dependencies of different projects. Hence, previously deployed Ionic projects should be removed from the test device to prevent side effects during testing. As a prerequisite to start and develop Ionic applications only knowledge in the typical Web development stack (HTML, JavaScript and CSS) is required. TypeScript as an extension of JavaScript and thus is easy to learn if JavaScript is already known. TypeScript provides additional benefits compared to JavaScript (especially type safety) – some extension have been adopted into ECMAScript-6 (such as classes, inheritance or generics) [24].

In addition, Ionic reuses Angular, which makes it easier to keep the code clean, separate concerns and speed up development of the application itself. The project structure in Ionic is logically well structured according to Web component architecture. Since Ionic relies on Web technologies, the user is free to choose the development environment [16]. The use of Cordova is another advantage, especially because it enables access to system specific functionality and device sensors. Furthermore, Cordova improves re-use of application components, since a single code base can be utilized for all platforms. However, since Ionic is based on Web-technologies and packaged into native wrapper applications, the performance is behind native applications. Former evaluations also indicated that the performance is behind Xamarin and React Native, especially for larger applications [20].

B. Xamarin

Xamarin projects can be set up in Visual Studio. With the use of C#, Xamarin is the best choice for developers, who also work conventionally with C#. Another advantage is the native UI [18]. Users will not recognize any difference to native applications. Xamarin offers to share a single code base between platforms, to develop application logic. Platform specific extensions may be integrated with a subproject feature of Xamarin. As for all cross-platform frameworks, problems may arise with third-party plugins (installed via NuGet). We recognized several issues with outdated plug-ins. In general, our experience has shown that new device and operating features of mobile devices had been adopted very fast by Xamarin. Hence, in most cases it framework based services can be used instead of third-party plugins. Regarding testing and debugging applications, the developers stated, that Xamarin would be the most convenient framework to use. This may be the case because of extended possibilities instantly provided by Visual Studio.

C. React Native

React Native is easy to set up as well. React Native is built upon React, and is also based on JavaScript. Applications developed in React Native interpreted directly and the design appears near to native. Interesting features include a well-designed live debugging. With Expo, React Native offers an open source toolchain to simplify deployment on test devices. Although this may result in some benefits, we observed that the apps that are generated by the Expo are structured differently than those set up by the console. Additionally, these apps have different access to native functions. Another disadvantage compared to the other frameworks is interface development. React utilizes a lot of specific HTML-Tags which we recognized as somewhat difficult to use and configure. This makes it more difficult to get started than with other frameworks, even if experience in Web technologies is preexistent.

D. Comparative Evaluation

To evaluate all frameworks comparatively and in an objective manner, we implemented a test application according the reference architecture (as defined in Section 3). In a second step, we measured the criteria defined at the beginning of Section 4, to reason about benefits and limitations of all frameworks.

Supported platforms: Ionic officially supports Android, iOS and since 2016 also UWP development, although the documentation is still very limited here. Xamarin offers full support for Android, iOS and UWP. Limited support is provided for MacOS. React Native supports iOS, Android and with a little extra effort also UWP applications. **Supported development environment**: Ionic applications can be developed on Windows, macOS and Linux. The development platform for Xamarin is Visual Studio for Windows and Xamarin Studio for macOS. React Native supports Windows, macOS and Linux. **Access to platform-specific functions**: Ionic provides access to iOS, Android, Microsoft and browser-based features. Platform-specific functions can be used via various Cordova plugins. With Xamarin, all platform-specific functions can be used in a similar fashion. However, Xamarin offers different possibilities to access platform specific functions. The fastest possibility is to install corresponding NuGet packages. A second option would be the definition of interfaces with platform or devices specific implementations and expose this shared code via the dependency service. Then there is also the possibility to use native libraries, for example written in pure Java for Android, via binding. While React-Native is JavaScript-based and many native functions are not supported, it is possible to include native SDKs and libraries. However, this requires specific code for Android (in Java) and for iOS (in Swift) which results in higher development efforts. In addition, these features are currently not mature enough.

Application look and feel: Ionic offers its own widgets for the UI. Navigation elements (e.g. back button) are provided in platform-specific style, so the differences to native apps are small. As already described in Section 3, the use of a GridView in Ionic is very simple. Xamarin creates completely native UI, thus the interface is familiar to the user. Xamarin also supports styling with themes and the interface is not different to native apps. Xamarin Android also supports material design. React Native uses specialized widgets. Setting up a GridView it is not as easy as in Ionic or Xamarin, CSS has to be used to achieve this layout. In general, Ionic and React Native ignore style guidelines of platforms partially and some widgets break them explicitly. For example, tabs in Android are at the top of the screen in native apps, while this is not the case in apps developed with

 ISBN: 978-1-61208-645-3

Ionic or React Native. Xamarin, in contrast, uses tabs as expected.

Application portability: Since Ionic represents a hybrid approach, portability of the source code is given and further supported through Cordova. Since Ionic modules are well-structured and based on Web technologies, they can be transferred to other Web frameworks. However, as many other frameworks, Ionic uses specific HTML tags that may not be supported in other frameworks, thus there is limited transferability of this module part. Since Xamarin separates application logic and UI related code, it offers the best portability and reuse of the logic. Furthermore, Visual Studio offers a portability analyzer to transfer the UI related parts as well. Of course, it has to be said that this is restricted to .Net and mono frameworks. The UI (defined by eXtended Application Markup Language (XAML)), could in principle be transformed into HTML or similar languages, which, however, requires further manual efforts in a second step. Similarly, React Native offers portability to different platforms. React-Native code is relatively easy to transfer to other frameworks that use JavaScript, HTML, and CSS. Comparable to Ionic, specialized HTML tags have to be XAML handled manually. However, since as React Native Logic, UI and CSS are typically implemented in a single file, this tends to be tedious.

Simplicity of development: Through a lot of documentation (tutorials, community discussion, API documentation, quick start and programming templates) a quick and efficient start in development of Ionic Apps is possible. Because of the short development lifecycle, confusion may occur through different version documents and some outdated plugins. Occasionally, the framework reveals unexpected behavior (some builds end up with broken apps, while a rebuild without code change is successful). We intend to examine this further. Currently we believe that this is related to generator issues. In principle, the development with Xamarin is fast as well, since the framework also possesses a very good documentation (tutorials, sample projects and a very precise API documentation). The programming language underneath (C#) also is very sophisticated and in our opinion much better then JavaScript. In terms of simplicity, Visual Studio or NuGet may pose a certain barrier for developers not used to it in the beginning. The entry into the development with React Native is comparable to Ionic. The use of the framework-specific UI elements is different from the other frameworks but does not impose an obstacle. The ability to see and debug all changes in real-time eases troubleshooting. A larger issue is related to external libraries and modules. Since many of these modules and libraries are not officially supported, regular maintenance and support is not guaranteed. In addition, we observed that the installation of node modules consumes much more time compared to Ionic.

Application performance: (100 test runs of the test app, with three test devices) The required start time of the Ionic test app is between 2s and 2.5s, while Xamarin requires 3 to 3.5s, and React Native 4 to 5s. The size of the Ionic app is 10 MB, while Xamarin requires 24 MB and React Native requires 11 MB. The time the Ionic app takes to retrieve the current location (with high signal strength of GPS) is approximately 0.2s, while Xamarin needs 3.2s and React Native 0.5s. Based on upon the evaluation criteria presented and measured above, the overall results are summarized in Table 1 (“++”=very good , “+”=good, “0”=neutral, “-“=poor, “--"=very poor).

TABLE 1: EVALUATION OF CROSS-PLATFORM FRAMEWORKS

Evaluation Criteria	**Ionic**	**Xamarin**	**React Native**
Supported platforms	+	++	+
Supported development platforms	++	0	++
Access to platform-specific functions	+	++	0
Application Look & Feel	+	++	+
Application Portability	0	0	0
Simplicity of development	++	+	0
Application performance	++	+	0

V. CONCLUSION AND OUTLOOK

Development of cross-platform applications is supported by different approaches: native development, transformation- or generator-based and interpreted. With the exception of native development, all approaches share in common that some layer of abstraction is introduced to encapsulate platform specific details. The latter enables platform-independent development in combination with platform-dependent deployment. In our case study, we examined three frameworks (Ionic, Xamarin and React-Native) in depth. In general, the performance of Ionic was reported to be the slowest of these three frameworks, with our test applications However, we measured Ionic to be the fastest of all three. Compared to other studies [4], we also observed performance enhancements of all evaluated cross-platform frameworks. This holds true for response and processing times as well as sensor access. The latter may be a result of framework improvements within the latest versions as well as improvements of the mobile device platforms. From a user perspective, there are no performance issues recognizable compared to native apps. All evaluated frameworks provide full access to system functionalities and sensors. Although new releases of operating systems provide new functionalities and sometimes also completely different API, the rate and speed of adoption in cross platform frameworks is quite high [25]. All cross-platform frameworks allow generic development for different operating systems, although there still exist limitations. As we discovered in the reference-app, sometimes apps are not completely portable, and still require platform specific adjustments. Furthermore, the re-use of components between different mobile applications is not yet supported. However, Ionic promises to allow the use of other components written in React, Vue and Angular in its next version [16].

Furthermore, long-term support is essential to reach a broad range of users and to enact support of up to date applications. With recent releases of the examined frameworks it could be observed, that backward compatibility to APIs of former releases was not given (e.g., between Angular1 to Angular2). This may be repeated with

 ISBN: 978-1-61208-645-3

new releases in the future. Thus, for mobile software engineering, it is questionable if standardized component development (e.g. Web component architecture) will be adopted and foster framework consolidation or if the proliferation of new programming languages and techniques will continue to split the market. Similar reasoning applies to backward compatibility of API. Consequently, further research questions regarding API issues arise. A major question will be, if it is possible to transfer code from current framework applications to new releases and preserve its functionality (even if the API changes). Therefore, as a next step, we intend to define a model-driven approach that tackles this issue. In this context, we plan to compare parser-based methods to transformation-based cross-platform approaches and derive API mappings.

REFERENCES

[1] Statista, “Internet of Things, Forecast, Number of networked devices worldwide by 2020,” Statista. [Online]. Available: https://de.statista.com/statistik/daten/studie/537093/umfrage/anzahl-der-vernetzten-geraete-im-internet-der-dinge-iot-weltweit/. [Accessed: 07-Mar-2018].

[2] Statista, “Number of smartphone users worldwide 2014-2020.” [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/. [Accessed: 15-May-2018].

[3] L. Corral, A. Janes, and T. Remencius, “Potential advantages and disadvantages of multiplatform development frameworks–a vision on mobile environments,” Procedia Comput. Sci., vol. 10, pp. 1202–1207, 2012.

[4] W. S. El-Kassas, B. A. Abdullah, A. H. Yousef, and A. M. Wahba, “Taxonomy of Cross-Platform Mobile Applications Development Approaches,” Ain Shams Eng. J., vol. 8, no. 2, pp. 163–190, Jun. 2017.

[5] Statista, “Android - Proportion of versions, Statistics,” Statista, Feb-2018. [Online]. Available: https://de.statista.com/statistik/daten/studie/180113/umfrage/anteil-der-verschiedenen-android-versionen-auf-geraeten-mit-android-os/. [Accessed: 07-Mar-2018].

[6] I. Dalmasso, S. K. Datta, C. Bonnet, and N. Nikaein, “Survey, comparison and evaluation of cross platform mobile application development tools,” in Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International, 2013, pp. 323–328.

[7] M. Lachgar and A. Abdali, “Decision Framework for Mobile Development Methods,” Int. J. Adv. Comput. Sci. Appl. IJACSA, vol. 8, no. 2, 2017.

[8] V. Ahti, S. Hyrynsalmi, and O. Nevalainen, “An Evaluation Framework for Cross-Platform Mobile App Development Tools: A case analysis of Adobe PhoneGap framework,” 2016.

[9] P. Smutný, “Mobile development tools and cross-platform solutions,” in Proceedings of the 2012 13th International Carpathian Control Conference, ICCC 2012, 2012.

[10] J. Perchat, M. Desertot, and S. Lecomte, “Component based framework to create mobile cross-platform applications,” Procedia Comput. Sci., vol. 19, pp. 1004–1011, 2013.

[11] M. E. Joorabchi, M. Ali, and A. Mesbah, “Detecting inconsistencies in multi-platform mobile apps,” in Software Reliability Engineering (ISSRE), 2015 IEEE 26th International Symposium on, 2015, pp. 450–460.

[12] R. Acerbis, A. Bongio, M. Brambilla, and S. Butti, “Model-Driven Development Based on OMG’s IFML with WebRatio Web and Mobile Platform,” in Engineering the Web in the Big Data Era, 2015, pp. 605–608.

[13] A. Holzinger, P. Treitler, and W. Slany, “Making Apps Useable on Multiple Different Mobile Platforms: On Interoperability for Business Application Development on Smartphones,” in Multidisciplinary Research and Practice for Information Systems, 2012, pp. 176–189.

[14] “Ionic vs React Native vs Xamarin 2018 Comparison | StackShare.” [Online]. Available: https://stackshare.io/stackups/ionic-vs-react-native-vs-xamarin. [Accessed: 07-Mar-2018].

[15] M. Ramos, M. T. Valente, R. Terra, and G. Santos, “AngularJS in the wild: A survey with 460 developers,” in Proceedings of the 7th International Workshop on Evaluation and Usability of Programming Languages and Tools, 2016, pp. 9–16.

[16] Ionic, “Build Amazing Native Apps and Progressive Web Apps with Ionic Framework and Angular,” Ionic Framework. [Online]. Available: https://ionicframework.com/. [Accessed: 07-Mar-2018].

[17] Apache Cordova, “Architectural overview of Cordova platform.” [Online]. Available: https://cordova.apache.org/docs/en/7.x/guide/overview/index.html. [Accessed: 09-Mar-2018].

[18] “Developer Center - Xamarin.” [Online]. Available: https://developer.xamarin.com/. [Accessed: 07-Mar-2018].

[19] Facebook, “React Native · A framework for building native apps using React.” [Online]. Available: https://facebook.github.io/react-native/index.html. [Accessed: 09-Mar-2018].

[20] A. Hevner and S. Chatterjee, “Design science research in information systems,” in Design research in information systems, Springer, 2010, pp. 9–22.

[21] M. E. Joorabchi, A. Mesbah, and P. Kruchten, “Real challenges in mobile app development,” in Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on, 2013, pp. 15–24.

[22] M. Willocx, J. Vossaert, and V. Naessens, “A Quantitative Assessment of Performance in Mobile App Development Tools,” in 2015 IEEE International Conference on Mobile Services, 2015, pp. 454–461.

[23] C. Rieger and T. A. Majchrzak, “Weighted Evaluation Framework for Cross-Platform App Development Approaches,” 2016, pp. 18–39.

[24] Microsoft, “TypeScript is a superset of JavaScript that compiles to clean JavaScript output,” 07-Jun-2018. [Online]. Available: https://github.com/Microsoft/TypeScript. [Accessed: 07-Mar-2018].

[25] M. Martinez and S. Lecomte, “Towards the quality improvement of cross-platform mobile applications,” ArXiv170106767 Cs, Jan. 2017.

A Scalable Vector Symbolic Architecture Approach for Decentralized Workflows

Chris Simpkin*, Ian Taylor*, Graham A. Bent†, Geeth de Mel† and Raghu K. Ganti‡
*Cardiff University, UK
Email: {SimpkinC,Taylorij1}@cardiff.ac.uk
†IBM Research, UK
Email: {Gbent,geeth.demel}@uk.ibm.com
‡IBM TJ Watson Research Center, USA
Email: rganti@us.ibm.com

***Abstract*—Vectors Symbolic Architectures (VSAs) are distributed representations that combine random patterns, representing atomic symbols across a hyper-dimensional vector space, into new symbolic vector representations that semantically represent the component vectors and their relationships. In this paper, we extend the VSA approach and apply it to decentralized workflows, capable of executing distributed compute nodes and their inter-dependencies. To achieve this goal, services must be discovered and orchestrated in a decentralized way with the minimum communication overhead whilst providing detailed information about the workflow - tasks, dependencies, location, metadata, and so on. To this end, we extended VSAs using a hierarchical vector chunking scheme that enables semantic matching at each level and provides scaling up to tens of thousands of services. We then show how VSAs can be used to encode complex workflows by building primitives that represent sequences (pipelines) and then extend this to support full Directed Acyclic Graphs (DAGs) and apply this to five well-known Pegasus scientific workflows to demonstrate the approach.**

Keywords*–*vector symbolic architectures; decentralized workflows; semantics; associative memory models.

I. Introduction

Workflows provide a robust means of describing applications consisting of control and data dependencies along with the logical reasoning necessary for distributed execution. For wired networks, there have been a wide variety of workflow systems developed [1]–[10]. A scientific workflow is a set of interrelated computational and data-handling tasks designed to achieve a specific goal. It is often used to automate processes, which are frequently executed, or to formalize and standardize processes. The majority of existing workflows rely on centralized management and therefore require a stable endpoint in order to deploy such a manager. In more dynamic settings, such as Mobile Ad Hoc Networks (MANETs) [11] where more collaborative applications (e.g., multi-user chats, or distributed analytics for coalitions [12], [13]) are needed, on demand workflows that are capable of spontaneously discovering multiple distributed services without central control are essential. The resulting distributed pathways are complex, and in some cases impossible to manage centrally because they are based on localized decisions, and operate in extremely transient environments.

In the current state-of-the-art, discovery of workflow steps is not dynamic; the exact services to be used for each workflow step must be specified in advance and, further, workflow orchestration is typically managed via a central point of control. Macker and Taylor [14] provides a mechanism that is decentralised, however, the specification of each workflow step, ip-address, connections and data, must be known in advance and is passed around between services via a JSON data block. Wittern et al. [15] present a graph based data model that aims to capture relationships between services and their usage detail for API ecosystems; the approach provides an interface that enables consumers to search for required services and service providers to obtain usage and compatibility statistics between the services and those of other providers as well as discovering new requirements. In contrast, we describe the use of VSAs to enable individual services to be self-describing and to learn contexts for which they are compatible. We describe a decentralised, multicast, environment where individual services can interrogate VSA encoded workflow messages, compute their compatibility to participate in a particular workflow step and be chosen if they are the best match. This is analogous to a group of humans listening to various work requests and deciding for themselves that they are capable and available to do a piece of work, i.e., all humans can understand the work request message and each human knows for themselves if they are available and capable of doing the the work. Naturally, more than one person may offer to do a particular job and we describe mechanisms for negotiating who gets to do the work.

However, applying VSAs to workflows requires several extensions and our contributions to this area can be summarized as follows:

- **Scaling Through Chunking:** To address scalability, we extend VSAs using a hierarchical vector chunking scheme that is capable of binding multiple levels of abstraction (workflow and sub-workflows/branches) into single vector. This approach scales to tens of thousands of vectors while maintaining semantic matching.
- **Encoding Workflows:** We employ our chunk encoding scheme to encode and decode very large sequences of services.
- **Representing Workflows Primitives:** We extend the encoding scheme to support Directed Acyclic Graph (DAG) workflows having one-to-many, many-to-many, and many-to-one connections.
- **Distributed Discovery and Orchestration:** We show how our VSA encoding scheme can be used for distributed discovery and orchestration of complex workflows. Workflow vectors are multicast to the network and participating services compute their own compatibility and offer themselves up for participation in the workflow.

We show that the result provides several desirable features and byproducts: it can encode workflows /sub-workflows that can be unbound on-the-fly and executed in a completely decentralized way; associated metadata can also be embedded

 ISBN: 978-1-61208-645-3

into the vector, e.g., security, configuration, etc.; the vector representation is extremely compact and self-contained and can be passed around using standard group transport protocols; and semantic comparisons or searches are scoped within a sub-group of services in a workflow, allowing scoped service matchmaking. We, then, apply the implementation to several Pegasus [16] workflows (Montage, CyberShake, Epigenomics, Inspiral Analysis, and SIPHT) and analyze the output.

The rest of the paper is structured as follows. In the next section, we provide some background into VSAs. In Section III, we outline the contributions we have made to scale VSAs and the extensions we applied for Workflows. In Section IV, we show how the VSA approach is applied to Linear Workflows and then to more complex DAG workflows. We discuss the resulting architecture and implementation in Section V, apply it to several Pegasus workflows in Section VI, and in Section VII, we conclude.

II. Vector Symbolic Architecture Overview

VSAs [17] are distributed representations that can be considered to sit somewhere between pure connectionist approaches and classic symbolic approaches, in that they employ *'atomic'* symbols to build complex representations of objects, like the classical symbolic approach to cognitive modeling and artificial intelligence research. However, the *atomic* symbols employed are random patterns of values spread over a hyper-dimensional vector space of dimension **N**. VSA symbols represent data and object features using random symbolic component vectors and combine these into vectors that semantically represent the component vectors and their relationships. VSA vectors are, therefore, said to be semantically self-describing [18]. *Atomic* vectors can be real valued like Plate's Holographic Reduced Representation (HRR) [19], typically having dimension $512 \leq N < 2048$, or binary vectors, such as Pentti Kanerva's Binary Spatter Codes (BSC) [20], typically having $N \geq 10,000$. For the work described here, we chose to build off Kanerva's BSCs, but most of the equations and operations listed and discussed should also be compatible with HRRs [21], too.

In BSCs, atomic symbols are assigned a random vector to represent an entity and due to the very high dimensionality employed, such *atomic* vector symbols are uncorrelated to each other with a high probability [20]. Higher level/complex concepts are built by combining *atomic* vectors using a *bundling* or *superposition* operation. A key advantage of this approach is that *superposition* involves no computationally expensive iterative learning of weights like traditional connectionist approaches [22]; rather, learning is achieved by direct combination of sub-features. An additional advantage is that VSA methods are completely deterministic and hence analysable and explainable, unlike traditional connectionist methods.

Superposition is the combination of sub-feature vectors into a same sized compound vector such that each vector element participates in the representation of many entities, and each entity is represented collectively by many elements [21]. Normalised Hamming Distance (HD) can be used to probe such a vector for its sub-features without unpacking or decoding the sub-features. If two high level concept vectors contain a number of similar sub-features, such vectors are said to be *semantically* similar, e.g., if we have three services:

$$Service1 = AudioIN_v + DeNoise_v + Convolution_v + Classify_v$$
$$Service2 = AudioIN_v + DeNoise_v + DFT_v + Classify_v$$
$$Service3 = AudioIN_v + LowPass_v + PowerSpec_v + Classify_v$$

where '+' is the superposition or bundling operator; then, comparing *Service1* with *Service2* will give a match since they have 3 common sub-features. Also, *Service1* and *Service2* will be more similar to each other than they are to *Service3*.

An issue arises, however, when using superposition to compare compound vectors in this way because such compound vectors behave as an unordered *bag* of features. Thus, if:

$$Service4 = AudioIN_v + DeNoise_v + Classify_v + ShutDownLine_v$$

then *Service4* would be equally similar to *Service1* as *Service2*, despite having a different output step. In order to resolve such issues VSAs employ a *binding* operator that allows feature values such as *DeNoise* and *Classify* to be associated with a particular *field name*, or *role*. This is analogous to how variable names are used in programming languages to associate values with a particular property, e.g., speed=3.

Services must agree upon a common method to assign atomic vectors for *roles* whereas feature vectors are usually compound vectors built up from lower level compound vectors and/or *atomic* vectors. When a *role* is bound to a value this results in a *role-filler* pair. Feature values such as *DeNoise* can be detected or extracted from the *role-filler* using an inverse binding operator. Bitwise XOR is used as both binding and unbinding with BSCs because it is its own inverse. In addition it is commutative and distributive over superposition ([20], page 147). It is also lossless, which means that both *roles* and *fillers* can be retrieved from a *role-filler* pair without any loss, e.g., using *'.'* as the bitwise XOR operator; if $\boldsymbol{Z = X.A}$ then $\boldsymbol{X.Z = X.(X.A) = X.X.A = A}$, since $\boldsymbol{X.X = 0}$, the zero vector. Similarly $\boldsymbol{A.Z = X}$. Due to the distributive property, the same method can be used to test for sub-feature vectors embedded in a compound vector as follows:

$$\boldsymbol{Z = X.A + Y.B} \tag{1}$$
$$\boldsymbol{X.Z = X.(X.A + Y.B) = X.X.A + X.Y.B} \tag{2}$$
$$\boldsymbol{X.Z = A + X.Y.B} \tag{3}$$

Examination of (3) reveals that vector **'A'** has been exposed, thus, if we perform $\boldsymbol{HD(X.Z, A)}$ we will get a match. The second term $\boldsymbol{'X.Y.B'}$ is considered noise because $\boldsymbol{'X.Y.B'}$ is not in our known 'vocabulary' of features/symbols. When a role and value (*filler*) are bound together this is equivalent to preforming a mapping or *permutation* of a vector value's elements within the hyper-dimensional space, so that the new vector produced is uncorrelated to both the role and filler vectors. For example, if $\boldsymbol{V = R.A}$ and $\boldsymbol{W = R.B}$ then $\boldsymbol{R}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ will have no similarity to $\boldsymbol{V}$ or $\boldsymbol{W}$. However, comparing $\boldsymbol{V}$ with $\boldsymbol{W}$ will produce the same match value to comparing $\boldsymbol{A}$ with $\boldsymbol{B}$. In other words, if $\boldsymbol{A}$ is closely similar to $\boldsymbol{B}$ then $\boldsymbol{V}$ will be closely similar to $\boldsymbol{W}$ because *binding* preserves distance within the hyper-space ([20, page 147]).

Thus, *binding* with *atomic* role vectors can be used as a method of *hiding* and *separating* values within a compound vector while maintaining the comparability between vectors.

This important property can be used to encode position and temporal information about sub-feature vectors within a compound vector. It also explains why we can state that $'X.Y.B'$ from (3) above, will not match to any known symbol, however, note that we can get back to B from $'X.Y.B'$; simply perform the appropriate xor's, $B = ((X.Y.B).X).Y$. We can now rephrase our *Service* description to differentiate its sub-features, i.e., we can reformulate $Service1$ to:

$$Service1 = Input_{rv}.\ AudioIN_v + Cleanup_{rv}.DeNoise_v + Process_{rv}.Convolution_v + Output_{rv}.Classify_v$$

This clearly resolves the incorrect matching between $Service1$ and $Service2$ with $Service4$. To test if $Service1$ uses $DeNoise_v$ as its cleanup step we perform:

$$HD(xor(Cleanup_{rv},\ Service1),\ DeNoise_v) \quad (4)$$

When using 10kbit vectors, if the result of (4) is less than 0.47 then the probability of $DeNoise_v$ being detected in error is less than 1 in 10^9 ([20, page 143]). If we have an audio signal we want to classify, we might multicast a request vector $Z = Input_{rv}.AudioIN_v + Output_{rv}.Classify_v$ which would cause listening services such as services 1, 2 and 3 to respond or become activated. We could further query the responding services to determine what type of cleanup and processing they do as per (4).

III. Extensions to VSAs for Workflows

The number of detectable sub-features that can be superimposed into a single vector is of limited capacity, 89 vectors for BSCs of dimension 10k [23], and this issue must be addressed in order to encode large workflows. Chunking is a bundling method that combines groups of vectors into a single compound vector which is then used as base for further bundling operations, recursively producing a hierarchical tree structure where each node in the tree is a compound vector, as shown in Figure 1. Various methods of recursive *chunking* have been described [19]–[21], [23]. However, such methods suffer from limitations when employed for multilevel recursion - some lose their semantic matching ability if any single term differs, others can not maintain separation of sub-features for higher level compound vectors when lower level chunks contain the same vectors. We addressed these issues and describe a novel recursive encoding scheme that provides semantic matching at each level by combining two different methods of permuting vectors.

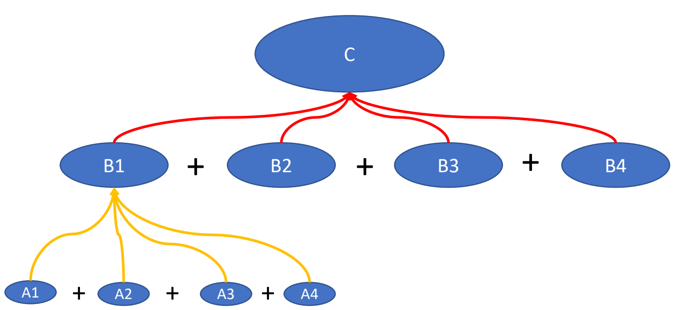


Figure 1. Workflow Chunk Tree, chunking proceeds from the bottom up.

In our scheme, the terminal nodes are worker services, the higher level nodes are concepts used to apply grouping to parts of workflow. The higher level nodes (known as *'clean-up memory'* [19], [20], [23]) are still services but they simply provide a proxy to the services to be *unbound* and executed, and thus are typically co-located with the first service of the sub-sequence, i.e., there is no network overhead. In a centralised system, *Clean-up memory* is typically implemented as an autoassociative memory. For our distributed workflow system, *clean-up memory* is implemented by the services themselves matching and resolving to their own vector representation.

Recchia et al., [24] point out that, for large random vectors, any mapping that permutes the elements can be used as a binding operator including cyclic-shift. The encoding scheme shown in (5) employs both XOR and cyclic-shift binding to enable recursive bindings capable of encoding many thousands of sub-features even when there are repetitions and similarities between sub-features:

$$Z_x = \sum_{i=1}^{x} Z_i^i . \prod_{j=0}^{i-1} p_j^0 + StopVec. \prod_{j=0}^{i} p_j^0 \quad (5)$$

Omitting $StopVec$ for readability, this expands to,

$$Z_x = p_0^0.Z_1^1 + p_0^0.p_1^0.Z_2^2 + p_0^0.p_1^0.p_2^0.Z_3^3 + ... \quad (6)$$

where

- $'.' = XOR$ and $'+' = BitwiseMajority\_Vote$.
- The exponentiation operator is redefined to mean cyclic-shift, $+ve$ exponents mean $Cshift\_right$, $-ve$ exponents mean $Cshift\_left$.
- Z_x is the next highest semantic *chunk* item containing a *superposition* of x sub-feature vectors. Z_x chunks can be combined using (5) into higher level chunks, e.g., Z_x might be $B1$, the superposition of $A1, A2, A3, ...$
- $Z_1, Z_2, Z_3, ...$ are sub-features being combined for the individual nodes in Figure 1. Each 'Z' is itself a compound vector representing a sub-workflow or a compound vector description for an individual service step.
- $p_0, p_1, p_2, ...$ are a set of known *atomic* role vectors used to define the current position/step in the workflow. The reason multiple $'p'$ vectors are XOR'ed together to define a single position within the workflow is to provide an iterative method for ordered activation of workflow steps during workflow execution, see (9).
- x is the, definable, *'chunk_size'*.
- $StopVec$ is a role vector that indicates to Z_x that all sub-feature/workflow steps have been executed.

A. Workflow execution

During workflow execution of a chunk tree similar to Figure 1 and encoded using (5), control first passes down the chunk tree, i.e., from $C \rightarrow B1 \rightarrow A1$, before traversing horizontally, $A1 \rightarrow A2 \rightarrow A3 \rightarrow A4 \rightarrow B1\_StopVec$. At this point $B1$ *'sees'* its own $StopVec$ and employs (9) to traverse horizontally at the next higher semantic level; see also flow arrows in Figure 2.

Referring to (6), an initiator or requester prepares the workflow, Z_x, for instantiation onto the network by performing an *unbind* operation, using (9), thus exposing the first workflow step, Z_1, as shown:

$$Z_1' = (T + p_0^0.Z_x)^{-1} \quad (7)$$

$$Z_1' = p_0^{-1}.T^{-1} + Z_1^0 + p_1^{-1}.Z_2^1 + p_1^{-1}.p_2^{-1}.Z_3^2 + ... \quad (8)$$

The $\boldsymbol{T}$ vector is a known *atomic* role vector used by $\boldsymbol{Z_x}$'s children to calculate their position within the sub-workflow. It is only bundled onto the workflow vector when an initiator or higher level node is requesting execution of its own sub-workflow, i.e., when traveling down the workflow.

In (8), note that all other $\boldsymbol{Z}$ vectors remain hidden because they are still permuted. Thus, listening services can only match to $\boldsymbol{Z_1}$. As control passes, horizontally, from $\boldsymbol{Z_1 \rightarrow Z_2 \rightarrow Z_3 \ldots}$ each active service uses the current permutation of the $\boldsymbol{T}$ vector to calculate its zero based position *'n'* within the currently active parent chunk vector. It can then activate the next workflow step in the chunk by repeating the *unbind* operation, generalized as:

$$\boldsymbol{Z'_{n+1} = \left(p_n^{-n}.Z'_n\right)^{-1}} \tag{9}$$

Hence, $\boldsymbol{Z'_2} = (\boldsymbol{p_1^{-1}.Z'_1})^{-1} = \boldsymbol{p_1^{-1}.p_0^{-2}.T^{-2}} + \boldsymbol{p_1^{-1}.Z_1^{-1}} + \boldsymbol{Z_2^0} + \boldsymbol{p_2^{-2}.Z_3^1} + \ldots$ Thus, execution proceeds in a completely decentralized manner whereby each node is activated when its preceding node, or parent, *unbinds* the currently active chunk vector and multicasts it to the network.

B. Local Arbitration

A major advantage of the VSA approach is the ability to find semantic matches because each service can extend beyond simple matches to include measures of real-time compute utility as well as policy. For certain scenarios, such as military coalition environments, there is a need to ensure multiple copies of services are distributed throughout the communications network. In order to decide which service is invoked, we employ a process of *local arbitration*, which is achieved as follows. Using terminology from (7) and (9), if the currently active service is $\boldsymbol{Z'_n}$, then before transmitting the next service request, it enters *match collecting mode* in order to arbitrate matches from all nodes that reply within a tunable window of time. After the interval expires, the highest ranking responder is selected and a *continue* message is broadcast by $\boldsymbol{Z'_n}$ identifying the winner. Since all communications are multicast, all services see all messages, and consequently the winning service continues and losing services discontinue. To reduce communication overhead further, matching services delay their response by an interval inversely proportional to their match value. Thus, better matches respond quicker. If a service *sees* an equal or higher match value before it has responded then it terminates without sending a reply.

C. Pre-provisioning and Learning to get ready

From (9) we see that each workflow step is exposed by iterative application of *'p'* vector permutations. Non-matching services can use this method to *peek* a vector enabling anticipatory behavior such as the pre-provisioning of a large data-set or changing a device's physical position, e.g., drones. Obviously, services can *peek* multiple steps into the future and could *learn* how early to start pre-provisioning. This ability to anticipate could be used to perform more complex, on-line, utility optimisation learning. For example, a drone, by monitoring multiple workflows may be able to understand that it will be needed in 10 minutes to perform a low priority task and in 15 minutes for a high priority task. Under these circumstances it may choose not to accept the low priority task.

IV. VSA Representation of Complex Workflows

As a test case and to compare to alternative approaches (e.g., [14]), we modeled each word of Shakespeare's play Hamlet as a service and applied hierarchical chunking to abstract into stanzas, scenes, and acts (see Figure 2). This approach tests the capability of the chunking scheme to encode serial and chunked workflows where the services at the lowest level are the 4620 unique words of the play. The semantic level above are the individual stanzas spoken by each character; the level above this are individual scenes of the play (e.g., A1S1, A1S2); next are the five acts, A1 to A5, and finally a single 10kbit vector semantically represents the whole play. The individual word services are distributed in a communications network and by multicasting the top level vector the whole play is performed in a distributed manner with 29770 word services being invoked in the correct order.

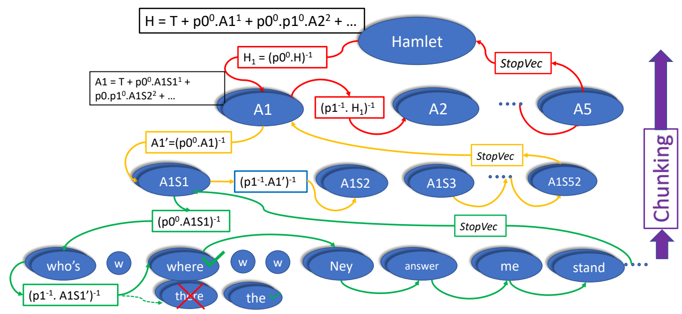


Figure 2. Hamlet as a serial workflow.

We employ a vector alphabet, a unique vector per character, and (5) to build a semantic vector description of each word-service in the test case. The idea is that each letter making up a word represents some feature of a service description, i.e., analogous to the different input/output/name/descriptions parts of a real world service. Thus, variable lengths of words and similarity of spellings represent a mix of different services of different complexity and functional compatibility. We can use this feature to find semantically similar service compositions when the best match composition is not available, i.e., we can find alternative words or stanzas, as shown in Figure 2, where the word *where* is selected as an alternate to *there*.

We note that this workflow is a linear sequence of services, next we show how such an approach can be extended to DAG workflows by employing three phases:

1) A recruitment phase where services are discovered, selected and uniquely named.
2) A connection phase where the selected services connect themselves together using the newly generated names.
3) An atomic *start* command indicates to the connected services that the workflow is fully composed and can be started.

Thus, in mathematical terms, using (6):

$$\begin{aligned}
\boldsymbol{WP} &= \boldsymbol{p0^0.(Recruit_{Nodes})^1 +} \\
&\qquad \boldsymbol{p0^0.p1^0.(Connect_{Nodes})^2 + p0^0.p1^0.p2^0.Start^3} \\
\boldsymbol{Recruit_{Nodes}} &= \boldsymbol{p0^0.Z_1^1 + p0^0.p1^0.Z_1^2 + \ldots p0^0.p1^0.p2^0.p3^0.Z_1^4} \\
&\boldsymbol{+p0^0.p1^0.p2^0..\,.p4^0.Z_2^5 \ldots + p0^0.p1^0.p2^0..\,.p9^0.Z_2^{10}} \\
&\boldsymbol{+p0^0.p1^0.p2^0..\,.p10^0.Z_3^{11} + p0^0.p1^0.p2^0..\,.p11^0.Z_4^{12}} \\
&\boldsymbol{+p0^0.p1^0.p2^0..\,.p12^0.Z_5^{13} + p0^0.p1^0.p2^0..\,.p15^0.Z_5^{16}} \\
&\boldsymbol{+p0^0.p1^0.p2^0..\,.p16^0.Z_6^{17} + p0^0.p1^0.p2^0..\,.p17^0.Z_7^{18}} \\
&\boldsymbol{+p0^0.p1^0.p2^0..\,.p18^0.Z_8^{19} + p0^0.p1^0.p2^0..\,.p19^0.Z_9^{20}}
\end{aligned}$$

 ISBN: 978-1-61208-645-3

$$\boldsymbol{Connect_{Nodes}} = (\boldsymbol{p0^0}.\mathbb{P}_1^1 + \boldsymbol{p0^0}.\boldsymbol{p_1^0}.\mathbb{C}_1^2) + (\boldsymbol{p0^0}.\boldsymbol{p1^0}.\boldsymbol{p2^0}.\mathbb{P}_2^3 + \boldsymbol{p0^0}.\boldsymbol{p1^0}.\boldsymbol{p2^0}.\boldsymbol{p3^0}.\mathbb{C}_2^4) + \ldots$$

where each $\boldsymbol{Z_n}$ in $\boldsymbol{Recruit_{Nodes}}$ is the semantic/compound vector representation of each service, built from the <job> entries found in the DAX. A generic description of each service was built from the service name and its description and used to build the workflow request vector. For individual instances of a service, e.g., *mDiffFit*, we additionally encode the instance's parameters and resource names to create similar but distinct service instances to, again, show that service discovery can be achieved when descriptions are not identical.

The resulting workflow, $\boldsymbol{WP}$, is a superposition representing the linear sequence of steps needed to discover, connect and initiate the workflow. Hence, execution of the workflow proceeds in a similar manner to that described in Section III-A, but with some additional workflow specific processing carried out by each selected node. The top-level vector, *WP* is prepared as per (7)

$$\boldsymbol{WP_1} = (\boldsymbol{T} + \boldsymbol{p_0^0}.\boldsymbol{WP})^{-1} = \boldsymbol{Recruit_{nodes}} + \boldsymbol{noise}$$

When multicast, this exposes and activates the $\boldsymbol{Recruit_{nodes}}$ service which, operating as a cleanup service, carries out the the same operation to initiate the recruitment phase.

$$\boldsymbol{Recruit'_{nodes}} = (\boldsymbol{T} + \boldsymbol{p_0^0}.\boldsymbol{Recruit_{nodes}})^{-1}$$
$$\boldsymbol{R'_1} = \boldsymbol{p_0^{-1}}.\boldsymbol{T^{-1}} + \boldsymbol{Z_1^0} + \boldsymbol{p_1^{-1}}.\boldsymbol{Z_1^1} + \boldsymbol{p_1^{-1}}.\boldsymbol{p_2^{-1}}.\boldsymbol{Z_1^2} + \ldots$$

where $\boldsymbol{Z_1}$ is a request for the first node in the DAG, an mProjectPP in the Montage DAG. This will be matched by all listening mProjectPPs. Acting as the local arbitrator, see Section III-B, the $\boldsymbol{Recruit_{nodes}}$ service multicasts its preferred match from the replies received. The newly discovered and activated service uses the current permutation of the $\boldsymbol{T}$ vector to calculate its position ($\boldsymbol{NODE_{id}^n}$) in the $\boldsymbol{Recruit_{nodes}}$ phase from which it can calculate its unique parent and child vector names to be used during the $\boldsymbol{Connect_{Nodes}}$ phase. Thus, the first mProjectPP, having position *p0* and being a $\boldsymbol{Z_1}$, calculates its parent and child names as,

$$\boldsymbol{\mathcal{P}_0} = \boldsymbol{Z_1^0} \cdot (\boldsymbol{NODE_{id}^0} \cdot \boldsymbol{ROLE\_parent})$$
$$\boldsymbol{\mathcal{C}_0} = \boldsymbol{Z_1^0} \cdot (\boldsymbol{NODE_{id}^0} \cdot \boldsymbol{ROLE\_child})$$

It then enters *Listening for Connections Mode* while, as the new *local arbitrator*, it also multicasts the next recruitment request by performing an unbind on its received vector $\boldsymbol{R'_1}$, thus $\boldsymbol{R'_2} = (\boldsymbol{p_1^{-1}}.\boldsymbol{Z'_1})^{-1} = \boldsymbol{p_1^{-1}}.\boldsymbol{p_0^{-2}}.\boldsymbol{T^{-2}} + \boldsymbol{p_1^{-1}}.\boldsymbol{Z_1^{-1}} + \boldsymbol{Z_1^0} + \boldsymbol{p_2^{-2}}.\boldsymbol{Z_1^1} + \ldots$ which will cause another mProjectPP to be selected and this decentralized process repeats until the last service to be recruited, the $\boldsymbol{Z_9}$, mjPeg, service unbinds and transmits the next vector, the $\boldsymbol{Recruit_{nodes}}$ *StopVec*. The $\boldsymbol{Recruit_{nodes}}$ cleanup service detects its stop vector, causing it to perform an unbind and multicast of $\boldsymbol{WP'}$ therby activating the $\boldsymbol{Connect_{nodes}}$ phase:

$$\boldsymbol{WP_2} = (\boldsymbol{T} + \boldsymbol{p_1^{-1}}.\boldsymbol{WP_1})^{-1} = \boldsymbol{Connect_{nodes}} + \boldsymbol{noise}$$

At this point, all recruited services are listening for connection request on their unique parent and child vectors. The activated $\boldsymbol{Connect_{nodes}}$ service, acting as a cleanup service, uses (7) to initiate and activate the first *parent* node of the $\boldsymbol{Connect_{nodes}}$ phase.

$$\boldsymbol{Connect'_{nodes}} = (\boldsymbol{T} + \boldsymbol{p_0^0}.\boldsymbol{Connect_{nodes}})^{-1}$$
$$\mathbb{P}'_1 = \boldsymbol{p_0^{-1}}.\boldsymbol{T^{-1}} + \mathbb{P}_1^0 + \boldsymbol{p_1^{-1}}.\mathbb{C}_1^1 + \boldsymbol{p_1^{-1}}.\boldsymbol{p_2^{-1}}.\mathbb{P}_2^2 + \ldots$$

When a service matches to its *parent* vector it simply performs the next unbind/multicast since in doing so it will activate its associated *child* service, automatically informing the child service of the location of its resources/output/ip-address.

When a service receives a multicast that matches to its *child* vector it can lookup the sender/parent's ip-address and send a unicast '*hello*' message to the parent, thus establishing the required connection before activating the next *parent* by performing a further unbind/multicast of the $\boldsymbol{Connect_{nodes}}$ vector. This process repeats until the final child request is processed causing the $\boldsymbol{Connect_{nodes}}$ service to detect its *StopVec* which, in turn, causes it to unbind and multicast the *StartVec* indicating to all nodes that the workflow has been fully constructed and processing can be started.

V. IMPLEMENTATION

Our VSA platform is implemented in `Python2` and has a modular architecture with several components that are capable of being reused as plugins to other systems.

The Workflow Importer component imports a Pegasus workflow description (DAX) file. This is an, XML format, multi-nested dictionary description of a workflow which details each service node and its input output resources. The Workflow Importer reads the DAX file into a python dictionary. It then parses the dictionary and extracts the *job* entries to create a list of vectors that represent each service node in the DAX, the *NodeVectors* list. Similarly, it traverses the *child* section of the DAX producing the *EdgeVectors* list, a paired list of vectors representing the parent (output) and child (input) connections of the workflow. The Workflow Importer passes *NodeVectors* and *EdgeVectors* to the VSA Creator.

The VSA Creator is used to bind the lists of vectors into a single vector, a reduced representation, of the workflow using chunking, see Section III. Chunking is performed bottom up so that higher level vectors are produced as needed. These are recursively rebound until the vector list is reduced to a single vector value. The *NodeVectors* list and the *EdgeVectors* list are combined separately producing two high level vectors, the $\boldsymbol{Recruit_{Nodes}}$ vector and the $\boldsymbol{Connect_{Nodes}}$ vector. The VSA Creator then binds these two vectors together with the *Start* vector into a single vector representing the entire workflow, the *WorkFlow* vector. This *WorkFlow* vector and all its associated sub-vectors are encapsulated in a *chunk tree* object as per Figure 1, which is then passed to the VSA executor.

The VSA Executor *flattens* the workflow by distributing copies of all non-terminal chunk vectors into the terminal (bottom level/worker) nodes. Non-terminal nodes are distributed to the first child of a parent node to decode the first vector in a higher level vector. For robustness, the VSA Executor can be made to distribute more than one copy of the cleanup objects into other terminal node objects.

The Logging Component collects metrics as the workflow runs to feed into external processors. Logging currently collects a trace of the nodes and edges that are being processed by the workflow.

ISBN: 978-1-61208-645-3

The Visualisation Component takes the log output and generates a DAG layout graph using Graphviz [25].

VI. Comparative Evaluation

For the evaluation, we imported five different DAX workflows generated using the Pegasus workflow generator [16]:

1) Montage (NASA/IPAC) stitches multiple input images together to create custom mosaics of the sky.
2) CyberShake (Southern Calfornia Earthquake Center) characterizes earthquake hazards in a region.
3) Epigenomics (USC Epigenome Center and Pegasus) automates various operations in genome sequence processing.
4) Inspiral Analysis (LIGO) generates and analyzes gravitational waveforms from data collected during the coalescing binary systems.
5) SIPHT (Harvard) automates the search for untranslated RNAs (sRNAs) for bacterial replicons in the NCBI database.

We ran the experiment on a MacBookPro11-4; Intel Core i7, 2.8 GHz; 4 cores; 16 GB memory using the CORE/EMANE network simulator using the Python toolkit discussed in the previous section in order to verify workflows could be loaded, encoded and then executed using the VSA format. We then recreate the workflows the VSAs have encoded to verify against the original. This proceeds as follows. Once imported, the DAX workflows are processed using the VSA creator to build the semantic vector workflow encoding, and apply the recruitment and connectivity phases to create service instances of the workflow jobs and interconnections. During the execution of the workflow using our simulator, we extract the metrics described in the previous section, which essentially contain a log of the run in the order of execution. This results in a set of nodes and edges being generated which we graph using Graphviz.

Figure 3 shows the resulting comparisons of the five workflows. The coloured images represent the Pegasus generated workflows and blue workflows show the VSA generated reconstruction of the workflows. Aside from the cosmetic difference, this demonstrates that all workflows were composed and correctly connected accurately in all cases.

VII. Conclusions

In this paper, we applied and demonstrated the viability of using VSA approach to encode workflows containing multiple coordinated sub-workflows in a way that allows the workflow logic to be unbound on-the-fly and executed in a completely decentralized manner. The Hamlet test-case demonstrated that we can use VSA service discovery to select alternate services on-the-fly. We anticipate that such an approach will lend itself well to edge networks where the transient nature of the mobile nodes will require such dynamic and decentralized control. In addition this test-case demonstrates that our encoding scheme is scalable, i.e., 30k individual services steps where successfully encoded and decoded in the correct order. The Pegasus test-case demonstrates the potential for encoding more complex multi-pathway workflows by encoding and decoding a number of Pegasus DAGs.

The local arbitration mechanism employed to choose the best matched services not only demonstrates a method that enables workflows to be orchestrated without a central point of control but can also be used to perform utility optimsation.

Using VSAs to enable services to become self-describing has a distinct advantage because of superposition. Conventional approaches could use multicast to transmit a bag of features across the network but each individual component feature within the bag would have to be examined and compared separately for a service to assess its compatibility to the request. In addition VSAs are robust to noise, they support mathematical inference and analogical mapping operations [19], [20] which could be used to learn similarities between vector symbols across coalitions and infer new workflows from previously seen workflows. For these reasons, our future work will therefore focus on such self-describing service compositions in order to realize the vision set out in [13].

Acknowledgements

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] M. Wieczorek, R. Prodan, and T. Fahringer, “Scheduling of scientific workflows in the askalon grid environment.” SIGMOD Record, vol. 34, no. 3, 2005, pp. 56–62.

[2] T. Fahringer et al., Workflows for e-Science. Springer, New York, 2007, ch. ASKALON: A Development and Grid Computing Environment for Scientific Workflows, pp. 143–166.

[3] Altintas et al., “Kepler: An Extensible System for Design and Execution of Scientific Workflows,” in 16th International Conference on Scientific and Statistical Database Management (SSDBM). IEEE Computer Society, New York, 2004, pp. 423–424.

[4] P. Kacsuk, “P-grade portal family for grid infrastructures,” Concurr. Comput. : Pract. Exper., vol. 23, March 2011, pp. 235–245.

[5] Deelman et al., “Pegasus: a Framework for Mapping Complex Scientific Workflows onto Distributed Systems,” Scientific Programming Journal, vol. 13, no. 3, 2005, pp. 219–237.

[6] Oinn at al., “Taverna: A Tool for the Composition and Enactment of Bioinformatics Workflows,” Bioinformatics, vol. 20, no. 17, November 2004, pp. 3045–3054.

[7] A. Harrison, I. Taylor, I. Wang, and M. Shields, “WS-RF Workflow in Triana,” International Journal of High Performance Computing Applications, vol. 22, no. 3, Aug. 2008, pp. 268–283.

[8] Barga et al., “The trident scientific workflow workbench,” in Proceedings of the 2008 Fourth IEEE International Conference on eScience. Washington, DC, USA: IEEE Computer Society, 2008, pp. 317–318.

[9] T. Glatard, J. Montagnat, D. Lingrand, and X. Pennec, “Flexible and efficient workflow deployment of data-intensive applications on grids with MOTEUR,” Int. J. High Perform. Comput. Appl., vol. 22, August 2008, pp. 347–360.

[10] B. Balis, “Increasing scientific workflow programming productivity with hyperflow,” in Proceedings of the 9th Workshop on Workflows in Support of Large-Scale Science, ser. WORKS ’14. Piscataway, NJ, USA: IEEE Press, 2014, pp. 59–69.

[11] S. Basagni, M. Conti, S. Giordano, and I. Stojmenović, Mobile Ad Hoc Networking: Edited by Stefano Basagni et al. IEEE, 2004.

[12] T. Pham, G. Cirincione, A. Swami, G. Pearson, and C. Williams, “Distributed analytics and information science,” in In IEEE International Conference on Information Fusion (Fusion), 2015.

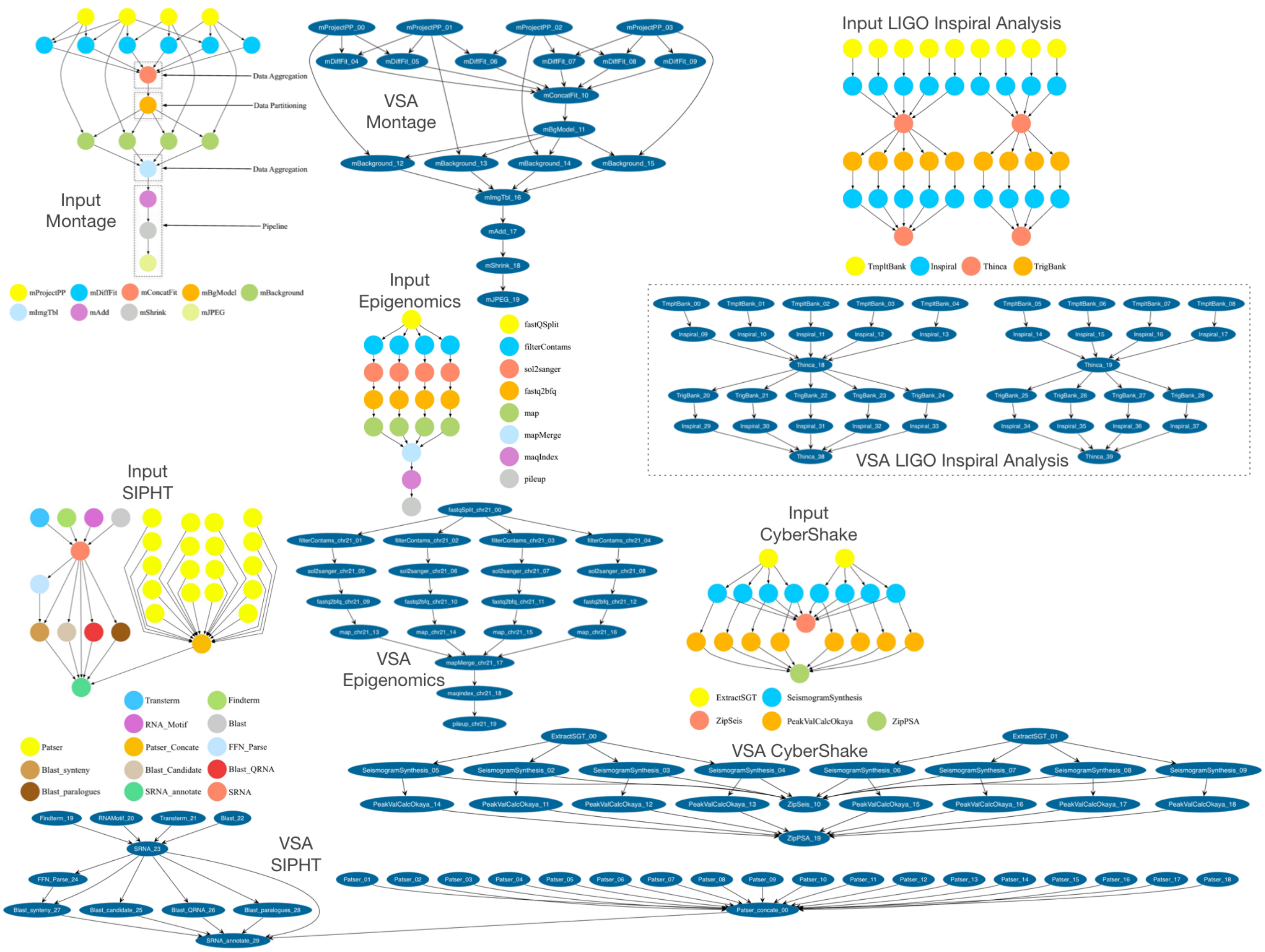


Figure 3. A comparison of five different DAX workflows as input and the VSA reconstructed workflows from post processing the semantic vector.

[13] D. Verma, G. Bent, and I. Taylor, "Towards a distributed federated brain architecture using cognitive iot devices," in 9th International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE 17), 2017.

[14] J. P. Macker and I. Taylor, "Orchestration and analysis of decentralized workflows within heterogeneous networking infrastructures," Future Generation Computer Systems, 2017.

[15] E. Wittern, J. Laredo, M. Vukovic, V. Muthusamy, and A. Slominski, "A graph-based data model for api ecosystem insights," in Web Services (ICWS), 2014 IEEE International Conference on. IEEE, 2014, pp. 41–48.

[16] "Workflow Generator Pegasus," https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator, [accessed on 30/05/2018].

[17] R. W. Gayler, "Vector symbolic architectures answer jackendoff's challenges for cognitive neuroscience," arXiv preprint cs/0412059, 2004.

[18] Eliasmith et al., "A large-scale model of the functioning brain," Science, vol. 338, no. 6111, 2012, pp. 1202–1205.

[19] T. A. Plate, Distributed representations and nested compositional structure. University of Toronto, Department of Computer Science, 1994.

[20] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors." Cognitive Computation, vol. 1, no. 2, 2009, pp. 139–159.

[21] T. A. Plate, Holographic Reduced Representation: Distributed Representation for Cognitive Structures. Stanford, CA, USA: CSLI Publications, 2003.

[22] J. L. McClelland, "Connectionist models james l. mcclelland & axel cleeremans in: T. byrne, a. cleeremans, & p. wilken (eds.), oxford companion to consciousness. New York: Oxford university press, 2009."

[23] D. Kleyko, "Pattern recognition with vector symbolic architectures," Ph.D. dissertation, Luleå tekniska universitet, 2016.

[24] G. Recchia, M. Sahlgren, P. Kanerva, and M. N. Jones, "Encoding sequential information in semantic space models: comparing holographic reduced representation and random permutation," Computational intelligence and neuroscience, vol. 2015, 2015, p. 58.

[25] "Graphviz - Graph Visualization Software," http://www.graphviz.org, [accessed on 30/05/2018].

Standardisation of Enterprise Architecture Development for Smart Cities

Zohreh Pourzolfaghar and Viviana Bastidas
School of Computing
Dublin City University (DCU)
Dublin, Ireland
zohreh.pourzolfaghar@dcu.ie
viviana.bastidasmelo2@mail.dcu.ie

Aleksas Mamkaitis and Markus Helfert
School of Computing
Dublin City University (DCU)
Dublin, Ireland
aleksas.mamkaitis2@mail.dcu.ie
markus.helfert@dcu.ie

***Abstract*— Managing the complexity of the Information and Communications Technology (ICT) services in smart cities raises a need to use enterprise architecture frameworks to solve the complexity issues. However, the majority of the existing enterprise architecture frameworks have been developed to address the concerns and issues of the stakeholders in their associated world. To address challenges including complexity, multi-stakeholders and the service-oriented nature of smart cities, this paper presents an enterprise architecture framework that can be used as a way to manage enterprise architectures in smart cities. This framework focuses on establishing contextual requirements and definitions for smart city systems and services. In contrast to other approaches, in this paper we focus on two important layers, i.e., context layer and service layer, as well as their relationships. The framework is valuable in developing smart services. It also contributes to the understanding of smart city enterprise architectures.**

Keywords-smart cities; enterprise architecture; smart services.

I. Introduction

Smart cities are complex systems, which use ICT services to improve citizens' quality of life. One of the current issues in smart cities is dealing with complexity of ICT services. For instance, challenges to connect various systems in smart cities [10] are due to the complexity of the smart cities. During the last decades, many Enterprise Architecture (EA) frameworks have been developed to manage complex information systems, processes and infrastructures in organisations and systems. Each of these enterprise architecture frameworks has been developed to address specific needs and concerns of the stakeholders and issues for their world [1]. According to a study comparing the frameworks by [1], we draw the conclusion that some of the well-known enterprise architecture frameworks have not considered the aspects which are critical for smart cities. Referring to the definitions for smart cities, the realisation of smartness is happening by providing services to the citizens [36]-[40][59]. Therefore, citizens (users) view is crucial for delivering effective services. Another example is related to the service life cycle. According to the comparison results by [1], the maintenance phase has been neglected for the majority of the frameworks. From the smart city perspective, a maintenance phase is crucial to deliver qualified and sustainable services to the citizens. Indeed, maintainability and sustainability are two of the quality factors [2] for smart cities.

According to the above mentioned discussion, an enterprise architecture framework for smart cities should consider smart city stakeholders and their concerns for improving the quality of life for citizens. For this purpose, we introduce a methodological framework to manage smart city enterprise architecture which can be used as a standardised approach to develop smart services. This framework contains two new layers, including the context and service layers, as well as the initial relationships between them. This paper aims to introduce the components and concepts to establish the new layers and relationships.

The remainder of the paper is organised as follows. In Section 2, five well-known enterprise architecture frameworks are reviewed to define the problem. In Section 3, the presented enterprise architecture for smart cities is introduced. In Section 4, it is explained how the evaluation of the created architecture is performed through simulation. In the discussion section, we argue that the presented enterprise architecture for smart cities can deal with smart city stakeholders and their concerns, leading to enhancement of the quality of the delivered services to the citizens.

II. Research method

This paper follows the design science research approach by [3] and [11] to present an enterprise architecture framework for smart cities. According to this research method, the first step is to define the problem. For this purpose, some well-known enterprise architecture frameworks are explored with regard to critical views for smart cities, i.e. citizens (user) view and their related phases in SDLC. The second step is to propose a solution for the recognised problem. To build up the solution artefact, this research follows the layered structure of an existing architecture by [7]. Then, relying on the related literature, the steps to construct the architectural layers and their relationships are elaborated. At last step, simulation is introduced as a method to assess the generated architecture. For this purpose, the created architecture for a service use-case in River city is evaluated and discussed.

III. Literature Review to Define the Problem

A. *Existing Enterprise Architecture Frameworks*

Designing an enterprise is a system engineering approach to determine the required capabilities for designing the organisation, processes, services, information, and technologies [12]. Architectures are created usually to manage and organise the complexity of systems. Architectures are utilised to construct blueprints of an enterprise for organising system components, e.g., interfaces, processes, services and much more [4]. Booch [5] stated that enterprise architecture is the way to architect and plan the enterprise to have the best human performance and output.

To describe and model various aspects of enterprises, researchers have proposed different approaches [7]. Most frameworks follow a process and integration aim [13]. Enterprise architectures usually consist of two main approaches: an enterprise architecture Framework together with an implementation methodology [14]. A common approach among prominent frameworks is the transition and implementation from strategic business objectives into the infrastructure and systems design. Enterprise architecture aims to support and enable this transition by providing ways to design concepts of an enterprise.

Usually frameworks use views and layers to describe architectural elements to manage complexity (e.g., process, service, and technology). Each of the views illustrates a different perspective meaningful to specific stakeholders. Layering decomposes a system into groups of related components whose processes provide services to subsequent layers. For instance, components like sensors at technology layer support an application layer by providing data to them.

Over the last decades, number of enterprise architecture frameworks including the Zachman Framework [6], Department of Defense Architecture Framework (DoDAF) (Chief Information Officer U.S. Department of Defense 2010), Federal Enterprise Architecture Framework (FEAF) [8] (Office of Management and Budget 2012), Treasury Enterprise Architecture Framework (TEAF), and The Open Group Architectural Framework (TOGAF) [9], have been developed. Common to these frameworks is reducing enterprises' complexities by considering disparate viewpoints and organising various aspects in ways that make an enterprise understandable. Despite of existing overlaps and similarities between these frameworks, each of them was designed to address specific needs and concerns of the stakeholders and issues for their world [1].

To specify the targeted concern and stakeholders of each framework, this study adopts the results of a comparison study by [1], whether the enterprise architecture frameworks encompass the entire software development life cycle (SDLC), as well as all stakeholders' views. The character 'Y' in the cells implies that the corresponding framework has provided details for the SDLC phase, or considered mentioned stakeholders' views. The adopted results are shown in Table I.

TABLE I. Comparison of the Enterprise Architecture Frameworks (Adopted from [1])

Framework	**SDLC Phases**					**Views/Perspectives**					
	Planning	***Analysis***	***Design***	***Implementation***	***Maintenance***	***Planner***	***Owner***	***Designer***	***Builder***	***Subcontractor***	***User***
Zachman	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y
DoDAF	Y	Y	Y	Y	-	Y	Y	Y	Y	-	-
FEAF	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	-
TEAF	Y	Y	Y	Y	-	Y	Y	Y	Y		-
TOGAF	-	Y	Y	Y	-	-	Y	Y		-	-

Reviewing Table I, empty (gray) cells unfold that critical aspects from smart city perspective have not been considered in the explored enterprise architecture frameworks. The first observation is that the user view has received the least attention, while the citizens/users are paramount in smart cities. According to the definition for smart services by [42], the ultimate goal for services in smart cities is to respond to the citizens' needs and concerns. Consequently, citizens (users) concerns are crucial in the enterprise architecture for smart cities.

The second observation is related to the maintenance phase which is of concern for authorities in smart cities, due to sustainability of the delivered services. As [2] stated sustainability and all other quality factors are crucial to realise smartness in the cities. Many standards (e.g., sensor security standards, [2]), and principles (e.g., open data), have been developed in smart cities, as the enablers of quality factors (e.g., security, confidentiality, [2]). In summary, it is concluded that some important aspects from smart city perspective cannot be addressed by the explored enterprise architecture frameworks. To address these challenges, this paper introduces an enterprise architecture in the following section.

IV. Designing Enterprise Architectures for Smart Cities

The overall view of the enterprise architecture for smart cities includes strategic and operational components as follows: vision and strategy management, portfolio management, service design, implementation and change management (See Figure 1).

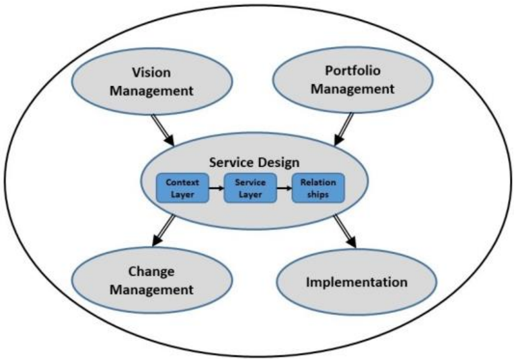


Figure 1. General overview of the framework to develop the presented enterprise architecture for smart cities

The focus of this paper is on the service design part. With this aim, this study follows [7] to put the architectural layers together. The presented layered architecture includes: Context layer, service layer, information system layer and technology layer (Figure 2).

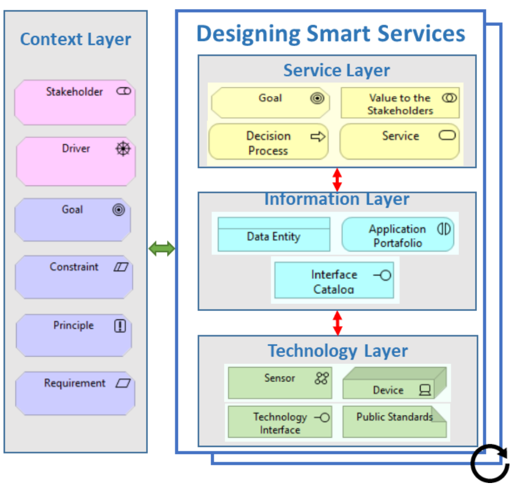


Figure 2. Overview of the architectural layers for design of services

The first two layers, i.e., context layer and service layer, have been initiated by this research for the first time. The context layer plays an important role to capture the contextual concerns from 'vision management' and transfer them to the service layer. In service layer, these requirements are considered to provide a detailed service description which will have relationships with information system layer and technology layer. A summary of the aims of focus of the presented layers is shown in Table II.

TABLE II. COMPARISON OF THE ENTERPRISE ARCHITECTURE FRAMEWORKS (ADOPTED FROM [1])

Architectural Layer	Aim and Focus
Context	To capture smart city context information about strategies, priorities and other critical aspects (e.g., stakeholders and their concerns), required to deliver effective services to the citizens.
Service	To define appropriate goals, scope and etc. for services with regard to the smart city requirements, concerns and priorities.
Information	To identify the data elements, data flows and the data interrelations required to support service function [4].
Technology	To support information function and the system/application function [4].

In the following three sub-section more details are provided for three main processes to capture smart city context information, to develop service description, and to relate the service description to the information layer.

A. Context Layer

To define the context layer, this study relies on two definitions for smart cities and smart services, and a taxonomy of the smart city requirements by [42]. The selected definitions are as below:

> *Smart cities are innovative cities which use ICT to facilitate daily activities of citizens to improve their quality of life* [58].
>
> *Smart services are the services which respond to the smart city stakeholders concerns and fulfil smart city quality factors* [42].

Regarding the above mentioned definitions, the context layer constitutes of the components to specify the stakeholders, their concerns, quality factors and their enablers/drivers. According to the taxonomy, the main stakeholders in smart cities are classified as follows: citizens, authorities, and service developers. Each of these stakeholders have their own specific concerns. Some examples of these concerns are: lower cost (for citizens), realisation smartness (for authorities), and more benefits and market share (for service developers). Likewise, the quality factors are defined as another component of the context layer. The quality factors (e.g., maintainability, interoperability, usability) should be fulfilled by the services in smart cities. The next two components are related to the enablers of the quality factors, i.e., standards (e.g., sensor security standards), and principles, (e.g., data principles). For any specific service, related standards and principles are required to be considered. Another component for context layer is the constraints (e.g., contractual constraints). This component is to avoid some consequences like low quality services, which are arising from some constraints like time and cost limitations. The last component for this layer is about documenting all the above mentioned initial considerations. The outcome of this process is a comprehensive collection of the considerations which have been defined based on the requirements and concerns for smart cities. All the above mentioned details are summarised in the form of a process. The defined activities for this process are as below:

- Specify smart city stakeholders, (considering the defined classes of stakeholders) to specify the target stakeholders for a specific service/initiative
- Specify stakeholders' concerns for various classes of stakeholders, to specify the target concern to be addressed by the service/initiative

 ISBN: 978-1-61208-645-3

- Specify the quality factors to be met by the service/initiative
- Specify related standards in the target domain for the service/initiative
- Specify related principles to enable the quality factors
- Specify the constraints (e.g., the budget for the project), to define appropriate goals for service/initiative
- Documenting all the considerations and specifications by the previous activities

All the before mentioned activities for this layer are utilised to define appropriate goal and scope for smart services/initiatives. In this way, alignment of the designed services with ultimate goal of the smart cities (which is improvement of quality of life) is ensured.

B. *Service Layer*

Service layer describes the "operant" resources of the service system, specifically service actors and their interactions. The types of service actors can be private persons, organizations, governments, and even countries, depend on the context [15] and the depth of service and service system analysis. This layer does not include resources, such as information or technology, and their function is to support actors and their interactions in the Service Layer.

Further, we present activities for the service system description. The aim of these activities is to extract and record contextual information for service modelling and design.

- Commence a service description including the necessary stakeholders who can provide the relevant service design information [16]
- Identify a function of the service [17] and [18] and its positioning within the Smart City domains [19]-[21]
- Articulate a use-case for the service [22]
- Clarify an exchange medium for service remuneration [23]-[27]
- List service actors [28] and types of their interactions
- Identify actor resources [29] based on each service actor contributions to the service
- Define an experience and Value proposition [27][30][31] that service is intending to provide
- Perform a PESTLE analysis to further understanding of the context within which the service will operate [32]-[34]
- Complete a service description by consolidating the information gathered from the domain [35].

The purpose of a service description is to provide a sufficient amount of information towards the Service System modelling and design.

C. *Relationships*

The service layer defines components such as city services, domains, stakeholders, locations, etc. to support the smart city goals and to facilitate and optimize intelligent decision making [42]-[44]. The information layer defines components such as applications, software services and data to support the automation or realization of city services [41][45][46]. The service layer and the information layer are physically and logically disconnected at the moment. Smart city architectures do not support the relationships across these architectural views. The identification of the architectural concepts and their relationships are essential for providing an alignment between these layers.

Smart city architectures must support the connection between service and information layers to ensure that citizens' needs are met by solutions provided by information technology [47]. However, in practice, smart city architectures do not present an application process to establish the relationships between the service and information layers. This makes smart city architectures fail to provide a foundation to guide the modelling of integrated and coherent models which meet the citizens' needs. The resulting models can fail to support the analysis, implementation, and maintenance of such city services.

To tackle this problem, this section defines a process for an alignment of service and information layers in smart city architectures. The proposed process aims to define the key activities regarding an identification of the relationships between architectural concepts of the service layer and the information layer. The key activities of a proposed process are defined as follows.

- Specify different smart city domains (e.g., health, education, mobility, environment, tourism, etc.) for each city service [48] and [49]
- Define the required data entities for each stakeholder (e.g., city authorities, citizens, communities, retailers, etc.) [50] and [51]
- Define the data entities which are provided or consumed by each city service [50] and [51]
- Define the digital tools, monitoring applications, application modules, or another deployable component to support each city service [44] and [52]
- Capture the location or place (e.g., roads, bridges, airports, tunnels, buildings, etc.) where software services operate [53] and [54]
- Define the required software services which support each city service [55] and [56]
- Specify a domain (e.g., health, education, mobility, environment, tourism, etc.) to which the software services belong [44][46][57]

The above mentioned process has been established to ensure appropriate relationships between the service and information layers. Also, it aims to help researchers and practitioners as follows. First, it helps organizations that need to design an architecture for smart cities to understand the issues associated with the relationships between these layers. Second, a formalisation of this process can help them to realise important advances in the design of more effective smart city architectures and make an industrial uptake of architectures research efforts easier. Finally, the proposed process supports the connection between the service and information layers to ensure that citizens' needs are met by city solutions provided by information technology.

V. EVALUATION

In this section, we will explain how the quality of a specific architecture can be evaluated by simulation. SimEvents software provides a simulation framework for analysing event-driven models to optimise performance characteristics such as latency, workload, conversion, and entity loss. Generators, switches, queues, servers and other predefined blocks enable us to model various important aspects of the system architecture such as processing delays, routing, prioritisation for communication and scheduling tasks.

A discrete-event system in a Simulink model is usually constructed from various blocks such as, generators, queues and servers. These blocks are used for producing and processing our entities, which represent discrete items of interest. Examples of entities are network packets in a communication system, customers in a restaurant, sensor readings or footfalls in an enterprise application. The motion and changes in entity attributes, corresponding to asynchronous events, update the system states such as length of a queue or entity service time in a server. In discrete-event systems, asynchronous discrete incidents (events) cause and affect the state transitions of the system.

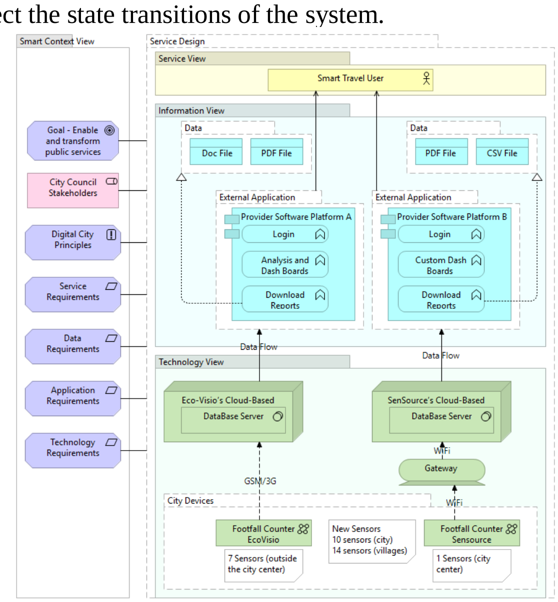
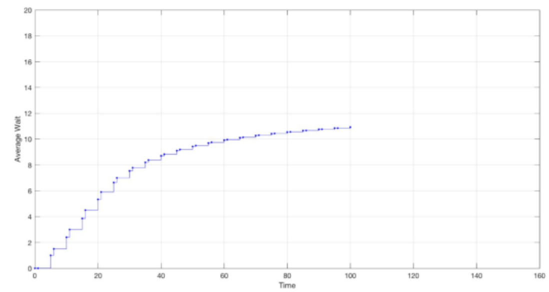


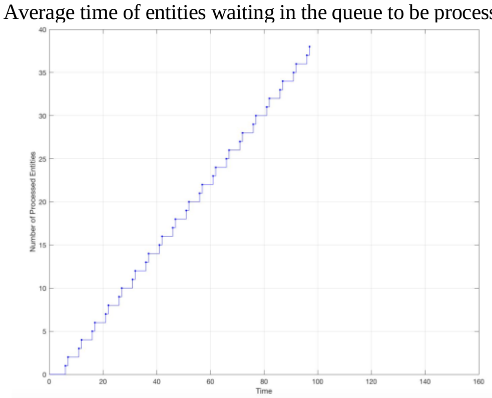
Figure 3. The real use-case from River city.

Figure 3 shows a very simple real use-case in the River city. The created architecture for the use-case stands on the presented enterprise architecture for smart cities. For this study's purpose, the right-side of the architecture shown in the figure is modelled. Specifically, we are interested in the effect of the Number of Processed Entities on the overall latency of the system which is measured by the number of entities reached the final destination.

For demonstration purposes, we will show two different results based on two different values of the "number of servers" property. For this purpose, the simulation is run for 100 time units, and is represented as the x axe in the following Figures 4, 5.

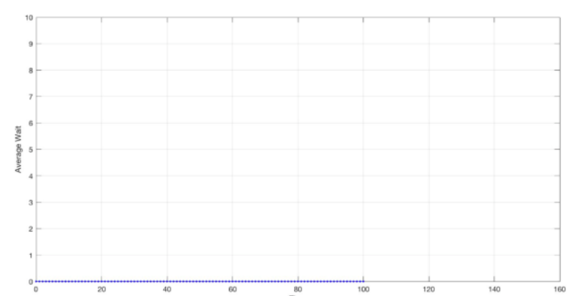


Average time of entities waiting in the queue to be processed

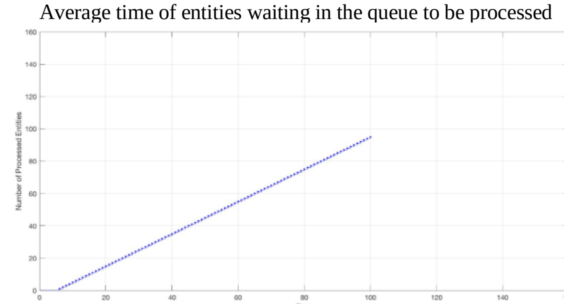


Number of processed entities reached the final destination

Figure 4. First Run (number of servers = 2).



Average time of entities waiting in the queue to be processed



Number of processed entities reached the final destination

Figure 5. Second Run (number of servers = 5).

It is obvious that an increase of the processing power (by increasing the number of servers from 2 to 5) leads to decrease of waiting time to almost zero. In addition, the final number of entities reached the final destination have become linear with the time. This simulation demonstrates how to evaluate the architectures which are created based on the presented smart city enterprise architecture.

VI. Discussion and Conclusion

Smart cities are complex systems which provide enormous ICT services to the citizens to improve their quality of life. Complexities in smart cities cause difficulties in management of provided services in terms of achieving smart city goals. Many years ago, enterprise architecture have been posed to solve complexity issues for organisations and systems. However, smart cities and organisations have different nature. Smart cities are service-oriented and organisations are business-oriented. Addressing the challenges arising from this difference, this paper presented the steps for designing new layers and relationships necessary for development of architectures for smart cities service. The first layer, i.e., context layer aims to capture smart city contextual information and transfer it to the service layer. The service layer provides information on service descriptions. This information is utilised by both, information and technology, layers. The relationships enable communications between the service layer and the information layer. The future study for this research will be defining other required relationships, e.g., between the service layer and the technology layer. The outcome of this research contributes to development of a reference architecture for smart cities.

Acknowledgment

This work was supported by the Science Foundation Ireland grant “13/RC/2094” and co-funded under the European Regional Development Fund through the Southern and Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

References

[1] L. Urbaczewski and S. Mrdalj, “A comparison of enterprise architecture frameworks,” Issues in Information Systems, vol. 7, no. 2, 2006, pp.18-23.

[2] G. Kakarontzas, L. Anthopoulos, D. Chatzakou and A. Vakali, “A conceptual enterprise architecture framework for smart cities: A survey based approach,” in e-Business (ICE-B), 11th International Conference on IEEE, 2014, pp. 47-54.

[3] K. Peffers, T. Tuunanen, M. A. Rothenberger and S. Chatterjee, “A design science research methodology for information systems research,” Journal of management information systems, vol. 24, no. 3, 2007, pp. 45-77.

[4] D. Minoli, Enterprise architecture A to Z: frameworks, business process modeling, SOA, and infrastructure technology, CRC press, 2008.

[5] G. Booch, “Enterprise architecture and technical architecture,” IEEE Software, vol. 27, no. 2, , 2010, pp. 96.

[6] J. A. Zachman, “A framework for information systems architecture,” IBM systems journal, vol. 26, no. 3, 1987, pp. 276-292.

[7] M. Meyer, M. Helfert and C. O’Brien, An analysis of enterprise architecture maturity frameworks. In Perspectives in Business Informatics Research. Berlin, Heidelberg: Springer, 2011, pp. 167-177.

[8] FEAF,The Chief Information Officers Council, Federal Enterprise Architecture Framework (FEAF) v1.1, 1999.

[9] The Open Group, The Open Group Architecture Framework TOGAF Version 9.1, 2011.

[10] Z. Pourzolfaghar and M. Helfert, Integration of Buildings Information with Live Data from IoT Devices, in Connected Environments for the Internet of Things. Cham: Springer, 2017, pp. 169-185.

[11] L. Ostrowski and M. Helfert, “Design science evaluation–example of experimental design,” Journal of Emerging Trends in Computing and Information Sciences, vol. 3, no. 9, 2012, pp. 253-262.

[12] R. E. Giachetti, Design of enterprise systems: Theory. architecture, and methods. Florida, 2010.

[13] J. W. Ross and G. Westerman, “Preparing for utility computing: The role of IT architecture and relationship management,” IBM Systems Journal, vol. 43, no. 1, 2004, pp. 5-19.

[14] B. D. Rouhani, M. N. R. Mahrin, F. Nikpay, R. B. Ahmad and P. Nikfard, “A systematic literature review on enterprise architecture implementation Methodologies,” Information and Software Technology, vol. 62, 2015, pp. 1-20.

[15] J. Spohrer, L. Anderson, N. Pass and T. Ager, “Service science and service-dominant logic,” In Otago Forum, vol. 2, no. 2, 2008, pp. 4-18.

[16] J. Cardoso, A. Barros, N. May and U. Kylau, “Towards a unified service description language for the internet of services: Requirements and first developments,” In Services Computing (SCC), IEEE International Conference, 2010, pp. 602-609.

[17] C. H. Lovelock, Developing and managing the customer-service function in the service sector. The Service Encounter: Managing Employee Customer Interaction in Service Business, Lexington, MA: Lexington Books, 1985, pp. 265-80.

[18] Q. Su, Z. Li, Y. T. Song and T. Chen, “Conceptualizing consumers' perceptions of e-commerce quality,” International Journal of Retail and Distribution Management, vol. 36, no. 5, 2008, pp. 360-374.

[19] T. Nam and T. A. Pardo, “Conceptualizing smart city with dimensions of technology, people, and institutions,” in Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times,ACM, 2011, pp. 282-291.

[20] P. Neirotti, A. De Marco, A. C. Cagliano, G. Mangano and F. Scorrano, “Current trends in Smart City initiatives: Some stylised facts,” Cities, vol. 38, 2014, pp. 25-36.

[21] R. Giffinger and N. Pichler-Milanović, Smart cities: Ranking of European medium-sized cities. Vienna: Centre of Regional Science, 2007.

[22] S. Adolph, A. Cockburn and P. Bramble, Patterns for effective use cases. Addison-Wesley Longman Publishing Co., Inc., 2002.

[23] R. G. Hansen, “A theory for the choice of exchange medium in mergers and acquisitions,” Journal of business, 1987, pp.75-95.

[24] R. G. Hansen, “A theory for the choice of exchange medium in the market for corporate control,” Journal of Business, vol. 60, no. 1, 1987, pp.75-95.

[25] S. Bignoux, “Short-term strategic alliances: a social exchange perspective,” Management decision, vol. 44, no. 5, 2006, pp. 615-627.

[26] M. Nowak and K. Sigmund, “A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game,” Nature, vol. 364, no. 6432, 1993, p. 56.

[27] B. Edvardsson, B. Tronvoll and T. Gruber, “Expanding understanding of service exchange and value co-creation: a social construction approach,” Journal of the Academy of Marketing Science, vol. 39, no. 2, 2011, pp. 327-339.

[28] J. M. Bryson, “What to do when stakeholders matter: stakeholder identification and analysis techniques,” Public management review, vol. 6, no. 1, 2004, pp. 21-53.

[29] S. Albert, B. E. Ashforth and J. E. Dutton, “Organizational identity and identification: Charting new waters and building new bridges,” Academy of management review, vol. 25, no. 1, 2000, pp. 13-17.

[30] P. Frow and A. Payne, “A stakeholder perspective of the value proposition concept,” European journal of marketing, vol. 45, no. 1/2, 2011, pp. 223-240.

[31] C. Gentile, N. Spiller and G. Noci, “How to sustain the customer experience:: An overview of experience components that co-create value with the customer,” European management journal, vol. 25, no. 5, 2007, pp. 395-410.

[32] G. C. Peng and M. Nunes, “Using PEST analysis as a tool for refining and focusing contexts for information systems research.”

[33] F. M. E. Team, PESTLE Analysis. Strategy Skills. Free management ebooks, 2013, p. 15.

[34] P. Newton, What is the PESTLE Analysis? 2014.

[35] A. Arsanjani, “Service-oriented modeling and architecture,” IBM developer works, vol. 1, 2004, p. 15.

[36] R. E. Hall, “The vision of a smart city,” in Proceedings of the 2nd International Life Extension Technology Workshop, Paris, France, 2000.

[37] L. Anavitarte and B. Tratz-Ryan, “Market Insight: Smart Cities” in Emerging Markets, Gartner, Stamford, CT, 2010, pp. 39-61.

[38] D. Ludlow and Z. Khan, “Participatory democracy and the governance of smart cities,” in Proceedings of 26th Annual AESOP Congress, Ankara, Turkey, 2012.

[39] J. Fenn and M. Raskino, Gartner's hype cycle special report for 2011, Stamford, CT: Gartner, 2011.

[40] R. Giffinger, C. Fertner, H. Kramar, R. Kalasek, N. Pichler-Milanovic and E. Meijers, “Smart cities–Ranking of European medium-sized cities,” Vienna: Centre of Regional Science, 2007.

[41] J. Massana, C. Pous, L. Burgas, J. Melendez and J. Colomer, “Identifying services for short-term load forecasting using data driven models in a Smart City platform,” Sustain. Cities Soc., vol. 28, 2017, pp. 108–117.

[42] Z. Pourzolfaghar and M. Helfert, “Taxonomy of Smart Elements for Designing Effective Services,” in Proceeding of 23rd American Conference of Information Systems (AMCIS), 2017, pp. 1–10.

[43] N. Mohamed, J. Al-Jaroodi, I. Jawhar, S. Lazarova-Molnar and S. Mahmoud, “SmartCityWare: A service-oriented middleware for cloud and fog enabled smart city services,” IEEE Access, vol. 5, 2017, pp. 17576–17588.

[44] A. Hefnawy, A. Bouras and C. Cherifi, “IoT for Smart City Services,” in Proceedings of the International Conference on Internet of things and Cloud Computing, 2016, p. 55.

[45] P. M. Singh, M. Van Sinderen and R. Wieringa, “Reference Architecture for Integration Platforms,” IEEE 21st Int. Enterp. Distrib. Object Comput. Conf., 2017, pp. 113–122.

[46] W. Rong, Z. Xiong, D. Cooper, C. Li and H. Sheng, “Smart city architecture: A technology guide for implementation and design challenges,” China Commun., vol. 11, no. 3, 2014, pp. 56–69.

[47] L. Anthopoulos, “Defining Smart City Architecture for Sustainability,” in Proceedings of 14th Electronic Government and 7th Electronic Participation Conference (IFIP2015), 2015, pp. 140-147.

[48] A. Gaur, B. Scotney, G. Parr and S. McClean, “Smart city architecture and its applications based on IoT,” Procedia Comput. Sci., vol. 52, no. 1, 2015, pp. 1089–1094.

[49] P. Neirotti, A. De Marco, A. C. Cagliano, G. Mangano and F. Scorrano, “Current trends in Smart City initiatives: Some stylised facts,” Cities, vol. 38, 2014, pp. 25–36.

[50] S. Consoli,V. Presutti, D. R. Recupero, A. G. Nuzzolese, S. Peroni and A. Gangemi, “Producing Linked Data for Smart Cities: The Case of Catania,” Big Data Res., vol. 7, 2017, pp. 1–15.

[51] A. Ilhan, R. Möhlmann and W. G. Stock, “A Community Architecture Framework for Smart Cities,” Citizen’s Right to Digit. City. Springer Singapore, 2015, pp. 231–252.

[52] M. Nitti, V. Pilloni, D. Giusto and V. Popescu, “IoT Architecture for a Sustainable Tourism Application in a Smart City Environment,” Mob. Inf. Syst. 2017.

[53] P. Nesi, C. Badii, P. Bellini, D. Cenni, G. Martelli and M. Paolucci, “Km4City Smart City API: An Integrated Support for Mobility Services,” in 2016 IEEE International Conference on Smart Computing, SMARTCOMP 2016, 2016.

[54] J. R. Gil-Garcia, T. A. Pardo and T. Nam, “What makes a city smart? Identifying core components and proposing an integrative and comprehensive conceptualization,” Inf. Polity, vol. 20, no. 1, 2015, pp. 61–87.

[55] E. F. Z. Santana, A. P. Chaves, M. A. Gerosa, F. Kon and D. Milojicic, “Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture,” arXiv Prepr. arXiv1609.08089, 2016.

[56] S. Clement, D. Mckee and J. Xu, “Service-Oriented Reference Architecture for Smart Cities,” in Service-Oriented System Engineering (SOSE), IEEE Symposium, 2017, pp. 81–85.

[57] N. Zakaria Bawany and J. A. Shamsi, “Smart City Architecture: Vision and Challenges,” Int. J. Adv. Comput. Sci. Appl., vol. 6, no. 11, 2015, pp. 246–255.

[58] ITU-T FG-SSC, “Smart sustainable cities: An analysis of definitions,” Technical report, ITU-T, 2014.

[59] M. L. Marsal-Llacuna and M. B. López-Ibáñez, “Smart urban planning: Designing urban land use from urban time use,” Journal of urban technology, vol. 21, no. 1, 2014, pp. 39-56.

Teamwork Behavior in Smart and Sustainable Cities Ecosystems

Panos Fitsilis
Dept. of Business Administration
University of Applied Sciences of Thessaly
Larissa, Greece
e-mail: pfitsilis@gmail.com

Paraskevi Tsoutsa
Dep. of Mathematics
University of Patras
Patras, Greece
e-mail: tsoutsa@teilar.gr

Leonidas Anthopoulos
Dept. of Business Administration
University of Applied Sciences of Thessaly
Larissa, Greece
e-mail: lanthopo@gmail.com

Omiros Ragos
Dep. of Mathematics
University of Patras
Patras, Greece
e-mail: ragos@math.upatras.gr

***Abstract*— According to market research, the number of smart cities is increasing rapidly. Information and Communication Technologies provide the smart infrastructure that is the foundation for all of the key themes related to a smart city, such as smart economy, smart people, smart governance, smart mobility, smart health, smart buildings, smart water, etc. As such, a smart city is constituted of various infrastructure components that form a complex system of systems, which is essential to collaborate effectively. Services play a central role in this vision of smart cities, as they are used as building blocks for effective collaboration, i.e., to achieve interoperability between heterogeneous parties of a business process and independence from the underlying infrastructure. In order to cope with the problem of complexity and the scalability in smart cities' systems, a solution is to provide autonomous, collaborating services that have situation awareness and they are able to adapt dynamically to the changing needs of the environment. In this research, we propose to model smart cities' services collaboration by using the role modeling approach enhanced by the introduction of service teamwork roles. The teamwork roles definition is inspired both by human and agent team working models. We contribute by determining the dominant teamwork roles that prevail during service group cooperation where the main goal of each role is to intervene during collaboration and "act as a connector" in order to keep the team of component services together and consistent with the goal of the group–team. The teamwork functionality is applied through the introduction of a new layer in the architecture of smart cities and is exploited to overcome some of the aforementioned problems.**

Keywords- Services; Smart Cities; Service Choreography; Role Modeling; Teamworking.

I. INTRODUCTION

According to market research, the number of smart cities is increasing rapidly. This growth is expected to continue for the next years, since the market of cities with population over 150,000 people is already 5,000 [1].

While there is no universal definition for the smart city, an often used definition is that of International Telecommunication Union (ITU) report in 2014, which is the following: "A smart sustainable city is an innovative city that uses Information and Communication Technologies (ICT) and other means to improve quality of life, efficiency of urban operation and competitiveness of services, while ensuring that it meets the needs of present and future generations with respect to economic, social and environmental aspects." [2].

ICT provides the smart infrastructure that is the foundation for all of the key themes related to a smart city, such as smart economy, smart people, smart governance, smart mobility, smart health, smart buildings, smart water, etc. As such, a city is constituted of various infrastructure components that form a system of systems [3].

According to Mayk and Madni [4], a System of Systems (SoS) is a collection of systems that were originally designed as separate systems for specific and different purposes but they have been brought together within the SoS umbrella, in our case a smart city umbrella, for creating new capabilities required for the mission of a smart city. A smart city as a system, is usually designed to accommodate various and diverse services, though not all services and their capabilities are defined precisely at least at the time of the initial deployment and, consequently, they are not included in the initial design. Further, since a smart city evolves dynamically, it could incorporate new enhanced and innovative services. This evolution includes continuous deployment of new services, reorganization of existing services and automatic development of novel more competitive services. Commonly, a smart city is characterized using terms, such as interoperable, synergistic, distributed, adaptable, inter-domain, reconfigurable, and heterogeneous. Actually, it is an ecosystem that evolves over its lifetime.

Furthermore, the key characteristics of SoS [5][6], which in our case are focused to a smart city should be a) operational independence of elements, b) managerial independence of elements c) evolutionary development, d) emergent behavior and geographic distribution. Although the above are quite important, however, another characteristic should be included, that of efficient collaboration, since the electronic habitants of smart cities are increasing rapidly

 ISBN: 978-1-61208-645-3

with the massive introduction of Internet of Things (IoT) technologies.

Collaboration of smart infrastructure will provide the cognitive framework that smart cities subsystems need in order to have the ability to cooperate transparently and autonomously for offering composite, complex and more efficient services. However, developing this collaborative cognitive framework is not straightforward, since according to Vlacheas et al. [7], while a wide set of predefined services and applications is available for employment in smart cities, there are technological barriers among objects when they are used across application domains.

Services are commonly considered for performing complex tasks, thus research has focused on the problem of the automatic service composition. An automatic and dynamic web service composition is a highly complex process and the proposed standards of related technology do not answer holistically the problems of web services discovery and composition yet [8]-[10]. Further, service composition and collaboration is limited to approaches that groups of services follow a plan for the composition taking into account only their functional behavior, and this cannot be characterized as collaboration. In many cases, services need to work together as a group to achieve common objectives, implying teamwork abilities, which are commonly used in case of groups of humans, agents or autonomous vehicles. Developing autonomously collaborative services capable of exhibiting teamwork behavior that would have situation awareness and adapt in the environment of a smart city is a challenge for our research. A service capable of exhibiting teamwork behavior is one that can effectively cooperate with multiple potential teammates on a set of collaborative tasks and that is able to intervene and catch errors or prevent emergent behaviors that will put in danger the overall team goal, i.e., the overall execution of a composite service.

Teamwork has become an important research field and its contribution to organizational performance has attracted attention of various research groups from several disciplines. In recent years, many scientists studied why humans succeed or fail in joint activities and a variety of models have been developed that follow social - psychological approaches for human team formation. Apart from human team working, another area where similar problems have been studied and such theories have been applied, relevant with this research, is the area of autonomous software agents and robots. As with humans, a group of autonomous software agents must accomplish given tasks by organizing themselves according to their individual characteristics and their teamwork behavior within the overall system.

Towards this direction, the problem we address in this paper is to exploit the existing architectures and augment them with a teamwork layer that may introduce the notion of teamwork collaboration within a set of services, agents and other systems that “live” in a smart city. More analytically, we present (i) the teamwork behaviors needed, and (ii) a teamwork software layer.

In order to do so, we exploit the role modeling approach and the definition of behaviors to create roles that, except their functional behavior, will also have teamwork behavior. Services that exist in a smart city could be modeled using this approach that directs the team of services, which are called to cooperate to reach a goal. The focus is on architecting cooperative teams of services that form composite services, where cooperative teams is meant in a stronger sense than composite services where it usually simply implies a set of bound services. This teamwork hypostasis of cooperation in services makes the concept of collaboration stronger.

The rest of this paper is organized as follows. Section II describes the background and related work in teamworking, in Section III we present a layer in the reference architecture for smart and sustainable cities, in Section IV we describe the proposed teamwork software layer, while the last section presents conclusion and ongoing work.

II. TEAMWORK THEORY

In many cases, smart cities’ environment is so complex that work is done in teams composed of members that are either humans, IoT devices, autonomous software agents where each of them is specialized in specific tasks. For example, teams of humans, vehicles and software services need to cooperate in various transportation scenarios [11], while humans, together with various smart cities cooperative agents offered by various providers implementing different protocols, need to collaborate. The success of a smart city use case depends on the team rather than the individual effort of each team member. In other words, a smart city is an ecosystem where, organization, businesses, human’s operators, users, IoT services and Cyber Physical Systems (CPS) are effortlessly connected while exhibiting teaming behavioral attitudes. In order for these entities to be effectively connected and collaborate as a team, models based on human team working literature were investigated and exploited.

Team working has been thoroughly studied by psychologists and human resource experts over the last decades. In early studies, some researchers have proposed models focused on specific characteristics that team members should have such as personality, functional expertise, competencies, goal orientations, etc. [12][13]. As the team research matured, research moved firmly from dealing with single characteristics that members should have to a variety of behaviors that members should expose [14][15]. Ultimately, in current research, member’s tasks and behaviors are often clustered into distinct roles within the system that are aligned with the expertise of each team member [16]. Increasingly, researchers propose that teammates, along with the operational tasks that they perform in a team, they have also to play some other teamwork role, such that of coordinator, contributor, idea generator, etc. Indeed, as it is witnessed by empirical studies, these approaches are effective in a variety of contexts, tasks and domains.

There were many attempts to classify roles and behaviors in the context of human team working. Among them, the most prominent models are: (i) Belbin's work in team roles [17], (ii) Parker' set of team player styles [18] and (iii) Margerison and McCann model, which defines eight different roles namely: (a) explorer–promoter, (b) assessor–developer, (c) thruster–organizer, (d) concluder–producer, (e) checker–inspector, (f) upholder–maintainer, (g) reporter–advisor, and (h) creator–innovator [19]. Each role is linked to predefined behavior and tasks. For example, "Creator–innovator" role is linked with forward thinking, new ideas and new ways of doing things. People playing this role in a team, come up with new strategies and different approaches to tasks, creating and experimenting with new ideas in order to handle various situations and challenges.

The various agent-based frameworks of team behavior proposed in the literature were also investigated to identify and analyze the different agent teamwork factors among various types of teams in related areas. Generally, agent based teamwork factors address different collaboration attributes, such as (i) how the team is organizing itself, e.g., by creating rules for collaboration and communication, (ii) how the team is forming its strategy for future direction, e.g., by planning and decision making and (iii) how the team work together to achieve synergy, e.g., by following rules of trust and engagement. Fifteen primary factors revealed from agent based models in the literature, which merit particular attention across different team tasks and group sizes. In the following, we present the three dimensions and how the teamwork factors are clustered [20]

a) Organizing factors, such as Collaboration (COL), Communication (COM) and Coordination (COO).

b) Strategy related factors, such as Planning (PLN), Learning (LRN), Decision making (DM), Evaluation (EVL), Teamwork policies (POL).

c) Synergy related factors, such as Ad hoc team setting (ADH), Autonomy (AUT), Delegation (DLG), Joint-intention (INT), Knowledge of teammates' capabilities (KCT), Knowledge sharing (KNL) and Trust (TRS).

The framework for the goals of this research paper is partially based on existing approaches in service modeling and teamwork behavior. However, according to our point of view, the primary objective for smart cities is to not only provide services coming from different smart systems and combine them through basic communication and collaboration. According to our opinion, the benefit comes through the ability to bind them in novel services that include the abilities of the composed services or systems and could implement the collective intelligence they have gained from the domain during their execution.

III. Smart and Sustainable Cities architectures

Smart cities are distributed-computing environments composed of a large number of software services. This architecture enables the continuous evolution of smart cities ecosystems. Therefore, a key quality of a smart city architecture is its ability to accommodate new services by automatically composing and executing as novel software services. This service composition process is critical, since in many cases there are interrelationship between city's core systems, given that these systems cannot work in isolation. On the contrary, it is quite common to operate in close collaboration, e.g., smart transport network relies on traffic management. Interconnecting these systems obviously improve their efficiency and intelligence. Numerous examples of such synergies exist e.g., smart water - smart energy, smart energy – smart buildings, etc.

Service composition in most cases is based on service orchestration, which is a basic concept of Service Oriented Architectures (SOA). Service orchestration is a centralized approach for composing services out of existing atomic software services. However, service orchestration is better suited on static environments, when there exists a coordinator, and plans are known in advance while minimal changes happen during the execution. Nevertheless, this is not the case in smart cities ecosystems, where changes are introduced frequently and the number and the type of offered services are not upfront defined. For such cases, service choreography is a preferred solution, since composition of services is done on a peer-to-peer fashion, leading to autonomously operating services. This need was early identified, thus choreographies were included in Business Process Model and Notation (BPMN 2.0) specifications [21].

As already mentioned, in the smart city case myriads of heterogeneous services operate independently. This trend will grow even more in the near future since smart services are transformed to sustainable cities that need to exhibit even richer behavior and functionality. According to ITU Telecommunication Standardization Sector (ITU-T) [22], the overall smart city architecture should provide support for Transportation services, E-government services, E-business services, Safety and emergency services, Smart health services, Tourism services, Education services, Smart buildings, Waste management services, Smart energy services, and Smart water services.

The only way to keep control of such complex systems is by developing services that are able to act and interact independently and on a demand basis. The development of smart cities' systems and applications demands software development environments that are able to support a number of functional and non-functional requirements. The functional requirements that need to be satisfied by a development environment, as they described in [23], are:

- Data management, which includes collection, storage, analysis, and visualization of city data.
- Applications Run-time support for facilitating deployment and integration of smart cities' applications.
- Sensor network data management and control.
- Data analytics functionality for analyzing massive volumes of data produces by a smart city.
- Service management according to SOA or other service management standards.

- Tools for conceptually defining smart cities models, organizations, etc.

Furthermore, a number of non-functional requirements need to be introduced to enrich the existing list, related mainly but not exclusively to the quality characteristics of the provided services. More specifically, some of the existing non-functional requirements are: interoperability, scalability, security, privacy, configurability, etc.

According to ITU-T [22], as shown in Figure 1, the Smart and Sustainable City (SSC) architecture should be layered and it consists of a) the sensing layer b) the network layer, c) the application layer and d) the Operation, Administration, Maintenance and Provisioning, and Security (OAM & P & Security). Many such frameworks have been presented in the literature, focusing on different architectural aspects. For example, in the work of Gaur, Scotney, Parr, and McClean a software framework is presented that includes a semantic layer, which enables exploitation of domain specific data based on the concepts and relationships between these concepts [24].

Anthopoulos and Fitsilis [25] explored various smart cities around the world and concluded that the architecture that is preferred by well-managed cases is the multi-tier architecture, which is applied in new, existing and smart planting cases, while it addresses both soft and hard infrastructure, and it considers natural environment and the evolving Internet-of-Things (IoT) in terms of sensor installation.

Obviously, architectures that in many cases were operational in the near past seem to be inadequate for the future and have deficiencies. For example, they are tuned for static service provisioning but not for dynamic service composition, they are controlled in most cases centrally and they do not allow ad-hoc collaboration between services (service choreographies) while smart city subsystems are interconnected and integrated but they do not exhibit intelligent behavior (situation awareness, adaptive behavior), etc. In order to overcome these deficiencies, in the sequel, we present the introduction of a new layer, which will provide teamwork functionality and could be utilized to solve some of the above-mentioned problems.

IV. TEAMWORK LAYER

In this paper, we do not present a complete architecture since specific functional requirements do not exist; they are generic and not analytical for a particular smart city implementation. What we discuss here, is best described as a conceptual application architecture layer, capturing the most prominent requirements for exhibiting an intelligent teamwork-collaborative behavior to smart cities' applications. However, this conceptual teamwork layer could be modified and transformed as necessary, to address all specific functional and quality requirements for specific smart city frameworks or projects. This teamwork layer, ideally, could be used with the role based modelling approach, for presenting the services/components that cooperates within the smart city ecosystem. According to this approach, for each smart city/domain a set of roles is defined. Each role is a set of behaviors. A role model of the domain contains all roles and their defined behaviors. The set of roles that each service acquires implies paths for the collaboration of the service with other services of the domain. By defining services as set of roles, it allows an abstraction and helps to capture other entities that might exist and cooperate in the domain to form heterogeneous teams, e.g., teams of services, agents and robots. This is also necessary since, although some services may have the same functional requirements, not all services can exhibit the same behavior concerning nonfunctional requirements. The role model and its application to model composite services is defined in [26].

Even though the role model is an abstraction that describes the patterns of interactions among a set of entities, our intention is to introduce in this paper specialized team work roles that intervene when particular behaviors extracted, for infusing team working behavior within smart and sustainable city services. By researching the related bibliography in human and agent team working, the

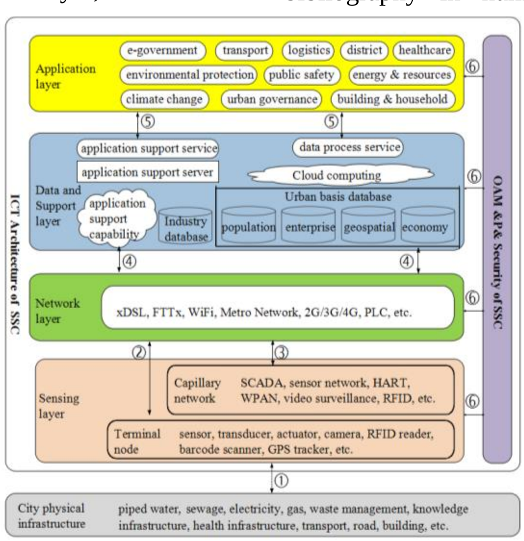


Figure 1. Smart and Sustainable City Architecture ITU-T [22]

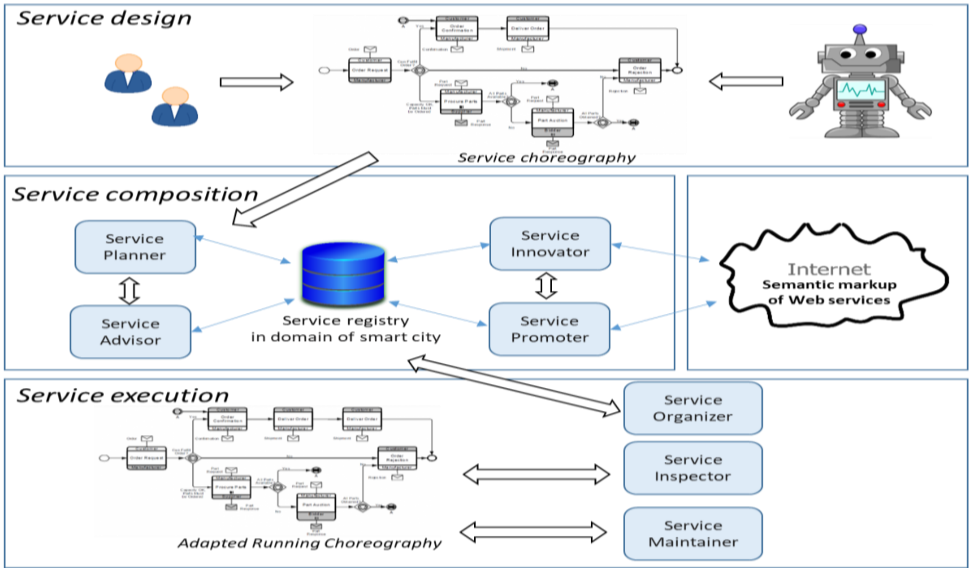


Figure 2. Smart and sustainable city service teamworking

following fundamental roles have emerged. The description of each role is minimal in the sense of reflecting solely the most substantial aspects of teamwork during the collaboration. Additional aspects occur in specific cases; these may be addressed by modifying the role and adapt it to the detailed behaviors.

Planner role. Complex smart city services need to be executed according to an overall plan. This role should be assigned to services having central function within different smart cities subsystems. For example, we should have one planner role for each different type of transportation service.

Organizer role is a role needed for organizing and monitoring the execution of different plans. It is possible that different services are assigned both the roles of planner and organizer, however these services should be strongly coupled (planners and organizers).

Similarly, the **inspector role** is the role that monitors the execution of the plan within the domain of a smart city. It is responsible to keep track of the progress, to inform organizer for possible delays and to trigger the planner when a new plan re-scheduling is required.

The **promoter role** is a role that enables interacting with different domains, e.g., by promoting services offered outside the domain of specific smart cities. A service having this role should be aware of the capabilities of all services of the domain. This is done with the aim to expand the offered services of a smart city or for looking about potential new services that are in demand.

Producer is a generic role that is assigned to all software services offering added value services to the citizens within the domain of a smart city.

Advisor role is needed for exploring new alternatives to develop the offered services within the domain of a smart city. Through history execution of services in the domain, it can reason and evaluate a set of services that will make the team formation. In a more advanced form, the advisor is based on semantic constructs such as domain ontologies in order to offer better reasoning using knowledge retrieval algorithms and provide recommendations of better quality.

Maintainer role. The main goal of the maintainer is to ensure that standards and processes of the domain are upheld and to maintain team functionality. It check for new updates for the system and keeps the configuration of the system.

Innovator role could be assigned to all experimental services of the domain, or to services running at a test system e.g., a landscape that have not been deployed yet.

As it is already mentioned, this team working model could be executed within a smart city domain to improve the collaboration of the composed systems. Services executed within the same domain will be aware of this fact, and will be able to share knowledge artifacts of the domain existing in various forms and formats. Further, through the role of the *advisor*, services are informed about other services that have similar behavior, their capacity, which of them are trusted, their functioning within the system in relation to the smart city's semantic model (e.g., smart city ontology) and the geospatial information of each service (e.g., which smart building is in the proximity of a smart vehicle). The satisfaction of these requirements would lead to the implementation of important operations within a team such as *delegation, common intention and knowledge sharing behavior*.

The implementation of team working services essentially follows the three-step implementation, which are: a) Service design phase, b) Service composition phase and c) Service execution phase. During the service design phase, a choreography could be designed using a business process modelling language, such as BPMN 2.0. The design is done after capturing the service requirements in collaboration with domain experts and users. Service composition could be done using BPMN 2.0 and after discovering the appropriate service instances needed. At this phase, after the *planer* provides a plan for the composition, the *advisor* role is needed in order to identify and evaluate the services published by providers within the domain. For describing the service plan, the behavior, the interactions, and messages exchanges need to be specified. This is implemented in the context of the *planner* role. The result of service composition

 ISBN: 978-1-61208-645-3

is the description of services to be executed and in this context the *inspector* role need to be activated. Meanwhile, service *innovator* is in collaboration with service *promoter* and according to the availability of services in service registry, they either promote existing services to be used or search for new available services that could be found from other providers through the internet. The *inspector* role in collaboration with the *organizer* role are the necessary components for running and monitoring the final choreography. This scenario is presented in Figure 2.

V. CONCLUSION AND FUTURE WORK

Collaborative processes performed by teams of services in a smart city environment require teamwork abilities. In this paper, we outline the principles for service teamwork roles, which are a fundamental building block for our proposed model of service teamworking in the context of a smart city. More analytically, we habilitate these roles with indicative teamwork behavior that services need as participants of an optimal team in order to underlie the required teamwork abilities and go beyond other approaches, by providing teamwork grounded service abstractions. Clearly, this is not sufficient and the analytical primitive behaviors should be designed that would be extracted from each team role according to the proposed role descriptions and the application domain. This is a step towards the definition of the architectural design and for creating a sophisticated smart city scenario where all defined roles will intervene and contribute, using real data from a smart city ecosystem. The benefits of applying this approach is that as smart cities are becoming even more complex systems we need to introduce concepts of intelligent collaboration and autonomous behavior that will allow such systems to evolve and to be managed easier.

REFERENCES

[1] T. Maddox, “Smart city technology market set to reach $775 billion by 2021,” [Online]. Available from https://www.techrepublic.com/article/smart-city-technology-market-set-to-reach-775-billion-by-2021/ June 2018.

[2] S. Kondepudi, et al., “Smart sustainable cities analysis of definitions,” The ITU-T Focus Group for Smart Sustainable Cities, 2014.

[3] L. Anthopoulos, “Understanding Smart Cities: A Tool for Smart Government or an Industrial Trick?,” vol. 22, Springer, 2017.

[4] I. Mayk and A. M. Madni, “The role of ontology in system-of-systems acquisition,” Tech. Rep., Santa Monica, CA: Intelligent Systems Technology, 2006.

[5] M. W. Maier, “Architecting principles for systems-of-systems,” Systems Engineering: The Journal of the International Council on Systems Engineering, vol. 1, no. 4, pp. 267–284, 1998.

[6] A. P. Sage and C. D. Cuppan, “On the systems engineering and management of systems of systems and federations of systems,” Information knowledge systems management, vol. 2, no. 4, pp. 325–345, 2001.

[7] P. Vlacheas, et al., “Enabling smart cities through a cognitive management framework for the internet of things,” IEEE communications magazine, vol. 51, no. 6, pp. 102–111, 2013.

[8] Q. Z. Sheng, et al., “Web services composition A decade’s overview,” Information Sci., vol. 280, pp. 218–238, 2014.

[9] A. Jula, E. Sundararajan, and Z. Othman, “Cloud computing service composition: A systematic review,” Expert Systems with Applications, vol. 41, no. 8, pp. 3809– 3824, 2014.

[10] D. Tosi and S. Morasca, “Supporting the semi-automatic semantic annotation of web services: a systematic literature review,” Information and Software Technology, vol. 61, pp. 16–32, 2015.

[11] T. Schwartz, et al., “Hybrid teams of humans, robots, and virtual agents in a production setting,” in Intelligent Environments (IE), 2016 12th International Conference on, pp. 234–237, IEEE, 2016.

[12] V. Rousseau, C. Aubé, and A. Savoie, “Teamwork behaviors: A review and an integration of frameworks,” Small group research, vol. 37, no. 5, pp. 540–570, 2006.

[13] E. Salas, N. J. Cooke, and M. A. Rosen, “On teams, teamwork, and team performance: Discoveries and developments,” Human factors, vol. 50, no. 3, pp. 540–547, 2008.

[14] S. W. Kozlowski, J. A. Grand, S. K. Baard, and M. Pearce, “Teams, teamwork, and team effectiveness: Implications for human systems integration,” APA handbook of human systems integration, pp. 555–571, 2015.

[15] J. M. Berlin, E. D. Carlström, and H. S. Sandberg, “Models of teamwork: ideal or not? a critical study of theoretical team models,” Team Performance Management: An International Journal, vol. 18, no. 5/6, pp. 328–340, 2012.

[16] J. E. Mathieu, S. I. Tannenbaum, M. R. Kukenberger, J. S. Donsbach, and G. M. Alliger, “Team role experience and orientation: A measure and tests of construct validity,” Group and Organization Management, vol. 40, no. 1, pp. 6–34, 2015.

[17] R. M. Belbin, “Team roles at work: A strategy for human resource management,” zitiert in: Teamarbeit und Teamentwicklung, p. 321, 1993.

[18] G. M. Parker, Team players and teamwork. Jossey-Bass San Francisco, CA, 2008.

[19] C. J. Margerison and D. McCann, Team management: Practical new approaches. Mercury Books, 1990.

[20] P. Tsoutsa, P. Fitsilis, and O. Ragos, “Teamwork Behavior: A Review to Interconnect Industry 4.0 Entities,” Handbook of Research on Emerging Developments in Industry 4.0, (in press), IGI Global 2017.

[21] T. Allweyer, BPMN 2.0: Introduction to the Standard for Business Process Modeling. BoD Books on Demand, 2016.

[22] ITU-T, “Setting the framework for an ICT architecture of a smart sustainable city,” Focus Group Technical Specifications, 2015.

[23] E. F. Z. Santana, A. P. Chaves, M. A. Gerosa, F. Kon, and D. S. Milojicic, “Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture,” ACM Computing Surveys (CSUR), vol. 50, no. 6, p. 78, 2017.

[24] A. Gaur, B. Scotney, G. Parr, and S. McClean, “Smart city architecture and its applications based on IoT," Procedia computer science, vol. 52, pp. 1089-1094, 2015.

[25] L. Anthopoulos and P. Fitsilis, “Exploring architectural and organizational features in smart cities," in Advanced Communication Technology (ICACT), 2014 16th International Conference on, pp. 190-195, IEEE, 2014.

[26] P. Tsoutsa, P. Fitsilis, and O. Ragos, “Role modeling of IoT services in industry domains," in Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences, pp. 290-295, ACM, 2017.

 ISBN: 978-1-61208-645-3

Gathering Insights on Citizen Participation
Towards Building an Instrument to Study Citizen-Centric Communities

Tobias Siebenlist, Agnes Mainka

Department of Information Science
Heinrich Heine University
Düsseldorf, Germany
Email: {tobias.siebenlist,agnes.mainka}@hhu.de

***Abstract*—Smart cities are not only about technological developments but also about committed citizens (which can be described as smart citizens) and projects that have been started by them. The opening up of administration and the idea of transparency will gradually lead to more data being published for free use. To generate added value from these raw data, however, they must be processed creatively. This paper addresses the development of an instrument to study communities concerned with open data and the development of tools, visualisations and projects of all kinds for their cities and their fellow citizens. The focus is on citizens, their backgrounds and their motivations for participation.**

Keywords–Citizen Participation; Community; Smart Citizen; Open Data; Smart City.

I. INTRODUCTION

When one thinks of smart cities [1], the focus is often on technical aspects, such as sensors or digitisation aspects in general. Administrations are also opening up, offering more transparency and new opportunities for citizens to participate. However, do smart cities also take their citizens on the way to become inhabitants of a smart city, maybe so-called smart citizens [2] and define their role? Moreover, what kind of people are these citizens who are already actively involved in this transition phase, want to participate and shape something with and for their city and fellow citizens.

Studies in the field of Smart Cities often refer to concrete projects within cities, to the transformation within the framework of digitisation as well as to fields such as sharing services or the opening of administrative activity (open government [3]). The area of Smart Cities is not limited to these factors and offers a wide range of aspects worth considering. An overview of the different elements is given in [4]. All these aspects are important, and some of them lead to more free and open data being available. This is particularly the case when you look at E-government and primarily Open Government and thus the right of citizens to inspect documents and processes. This right involves the publication of open government data, which should be available in machine-readable form as raw data. The data sets are often made available by cities and municipalities on open data portals and can be easily obtained. However, someone has to work with this data, develop something from it, create added value. Since these are data from specific cities, the citizens of these cities, in particular, can gain insight into what is happening in their city. These citizens, who have further knowledge of the city in which they live, are therefore particularly suitable for further processing. Depending on the type of further processing, various skills are required. In this article, the focus is therefore on the interested and committed citizens who join forces to do something for their city, their fellow citizens and thus also themselves. Therefore, the first steps on to a research tool are described to develop an understanding of who these committed citizens are and what motivation drives them to get involved. The Open Knowledge Labs, initiated by the Open Knowledge Foundation, are used as a starting point, as interested people develop tools or visualisations for their city without too particular reference to specific topics. Initially, the Open Knowledge Labs in Germany will be included, but the tool to be developed should not be explicitly limited to this, but can also be used for other countries and similar communities.

The structure of this paper is as follows. In Section 2, the Open Knowledge Labs in Germany will be introduced. The related work is mentioned in Section 3. In Section 4, the methodology towards creating the instrument is described. Finally, in Section 5, conclusions are drawn, and future work is mentioned.

II. OPEN KNOWLEDGE LABS IN GERMANY

It is difficult to identify why individual citizens take part in participatory processes and how (e.g., using a digital service provided by the city via the web or an app). Sometimes an app is only tested or used sporadically, or a single request to a specific service is made. On the other hand, groups that meet regularly to develop their city or develop something for their city often have established a long-term interest and motivation for this topic. These communities can then be examined regarding their composition, how the individual members find their way into the community and what motivates them to deal with such topics in their free time. However, these are only a few aspects that affect the participants. Examples of such groups are the Open Knowledge Labs (OK Labs), which were initiated by the Open Knowledge Foundation. The focus is initially on Germany, where the Code for Germany project was started. The project aims to network developers, designers and people interested in open data, who meet regularly to exchange ideas and work together. The aim is to develop and promote projects and applications around open data concerning the own city. All people are welcome, regardless of their professional or personal background. The term *Code* is not meant to be restrictive: it is not only about programming, but it is also a desired ability. Currently, there are 25 Open Knowledge Labs in different parts of Germany. In the context of a future study, the instrument will be used at all 25 labs in Germany to obtain an overview of the composition, topics, motivations and organisation. In the following, the instrument is to be made adaptable and standardised concerning the aspects of the investigation so that it can also be used in other countries. The consideration of local peculiarities is explicitly provided for. The study concerns the participants, the organisers and the environment of the labs. The focus is on the people, their backgrounds and motivations in this citizen-centred approach. The individual dimensions of the investigations are explained in more detail in the section *Methodology*.

III. Related Work

The study of communities dedicated to a topic is not fundamentally new and has already been carried out for different types of communities and taking into account various aspects. For example, an empirical study on the participation in free and open source (online) communities shows that ”70% of the surveyed members, stated that meeting and talking to people with similar interests was one of their main reasons for their participation.” [5] The basic motivation has been described in this paper as to perform some function and to do something. Whereas *social exchange* seems to be important, *problem solving* was the reason with the most votes. As the instrument will use a citizen-centric approach, one central aspect is the motivations why and how citizens engage in such communities and projects. As open source communities seem to have a similar approach (despite they often exist in the form of an online community instead of local groups that meet regularly), research on these communities can be helpful to build a research instrument for Open Knowledge Labs. The motivation for participation in open source software communities has been investigated by Ke & Zhang [6]. They relied on different motivational theories and referred to different types of motivations in the context of open source software communities regarding the participational aspect. From that, they composed their research model which focuses ”... the interaction effects of external motivation and personality traits, and of identification motivation and personality traits.” [6] This research model can be a basis for the studying the motivational aspects in our instrument. Their results based on a study carried out in specific open source communities, collected using an online survey.

In contrast to these and other such studies, the Open Knowledge Labs have a distinctive characteristic: the meeting does not take place because of a fixed, tangible goal, but because of the interest and motivation to do something for one’s environment. It is not clear what kind of results are obtained, what tools are used or how the results are communicated and disseminated. There are presumably very heterogeneous groups here, which are dedicated to different projects and which are connected only by the common motivation for the improvement of the city. Regarding improving the city, however, many aspects, such as the environment, transport, communication, are possible. The intended instrument should, therefore, be as general and broad as possible, so that many factors can be included in the inventory and a multi-perspective overview emerges as a result, which can be used for further investigations. Through such a common interest, a parallel can be drawn with other types of communities: Communities of Interest [7] or Communities of Practice [8]. In a study on the classification of communities, Herranz et al. [9] highlighted the differences between Communities of Interest and Communities of Practice. Based on this study, a categorisation of Open Knowledge Labs as Communities of Interest seems more appropriate, as they have a complex character, can change in personnel at any time and pursue common goals that are not tied to specific methods or tools. The further development of the instrument can build on this classification. From the research results obtained, an evidence-based decision can subsequently be made regarding categorisation of the individual Open Knowledge Labs.

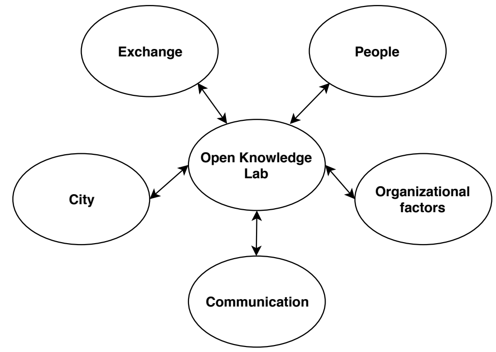


Figure 1. Core Dimensions of the Research Instrument

IV. Methodology

To get a complete picture of these communities and their members, a research tool is being developed to investigate these and other aspects. The dimensions of this instrument are shown in Figure 1.

The five core dimensions (*People*, *Organisational factors*, *Communication*, *City*, and *Exchange*) build up the basis for the investigation instrument. Every dimension carries further aspects with it, which need to be further developed and refined. The aspects of the individual dimensions are described in more detail below.

A. People

Since the approach of the investigation instrument is aimed in particular at the participating citizens and their motivation as well as their professional and personal characteristics, this aspect comes first. In addition to general demographic data such as age, gender and the number of participants, the professional background (training and current employment) and in particular the motivational factors for participating in such a community are collected. The evaluation also distinguishes between lab leads, regular participants and sporadic participants. Also, the wishes and expectations of participants and potentially interested parties are asked. If possible, the reasons why people have left an Open Knowledge Lab should also be collected.

B. Organizational factors

The organisational factors include aspects that deal with the organisation of an Open Knowledge Lab. This consists of the organisational and management structure, as well as local and temporal factors. Is there a fixed location for the meetings and a fixed time frame? How often do meetings take place and is there the possibility of spontaneous changes regarding space and the extension of meetings? Besides, the financing

 ISBN: 978-1-61208-645-3

of projects and the legal status of Open Knowledge Labs are examined.

C. Communication

The fact that a community exists like an Open Knowledge Lab must first get around. This dimension deals with various aspects of communication. How are meetings announced and where can I find out more about the idea, the current projects and in general get more information on the Open Knowledge Lab? It is examined whether a website exists, which social media channels and community tools are used for what purpose and in what frequency, and generally how the public relations work of the individual labs is carried out. Among other things, it is essential to know how new members can find their way to an Open Knowledge Lab and how completed projects can be prominently presented, which in turn can arouse the interest of other interested citizens and representatives of the city.

D. City

The involvement of the city or people from the administration in the activities of the Open Knowledge Lab is an important aspect when it comes to implementing projects to improve the city. Here, it is examined whether the city or municipality operates an Open data portal or there are persons in the administration who are responsible for this area. Also, the connection to the administration is considered concerning whether an exchange and, if applicable, participation in Open Knowledge Lab meetings have been realised and what experiences have been made with them. Furthermore, the involvement in urban decision-making processes, the transparency of the administration as well as the general state of the city on the way to a smart city is considered. This makes it possible to assess the Community's activities with the development phase of the city. It is also of interest what experience the citizens have already had with contact with the administration, what paths they prefer and what possibilities can be created here in joint projects. Studies like this from Reddick & Anthopoulus [10] can be consulted to investigate interaction with administrations further. Another aspect is to investigate whether institutions (such as universities) or associations (such as digital innovation hubs or business incubators) exist in the respective city that promote the presence and development of smart citizens.

E. Exchange

Open Knowledge Labs are usually not the only one of communities working on digital projects. For projects from the Open Source environment, as well as other community projects (e.g., Linux, OpenStreetMap, etc.) there are often local user groups that come together for similar reasons, but often with more concrete projects and ideas. This dimension investigates whether there is cooperation or contact with other communities in the city or the surrounding area and to what extent this can influence the work. Besides, there is a link to the organisational factors, for example, the question of whether several communities share a meeting point.

F. Realization

The instrument will be used in surveys and interviews. These methods are used to collect data during personal visits to all Open Knowledge Labs. Alternatively, the data can also be obtained online and gathered again on a regular basis.

V. Conclusion and Future Work

The presented instrument should enable an inventory of data on people and their motivation to participate in topic-specific groups in participatory activities to improve the city. Before a first concrete application, the individual dimensions and the aspects contained therein are rechecked and supplemented or revised. Also, tests are carried out in which the aspects are processed within the framework of specific surveys or interview guidelines. The contacts to some labs already exist, and the membership of the authors creates direct access to the persons. The results will provide a first insight into the structures, composition and working methods of Open Knowledge Labs, which are not yet available and have not yet been examined in their entirety. From this data collection, measures to optimise the work and possible positive and negative influences can be discovered. In further steps, the data can be used for analytical purposes and international comparisons, for example, if corresponding data collections also take place in other countries. Similarly, this can lead to further dialogue between citizens and the city administration and strengthen mutual understanding if both sides approach and work together transparently.

References

[1] R. G. Hollands, "Will the real smart city please stand up? intelligent, progressive or entrepreneurial?" City, vol. 12, no. 3, 2008, pp. 303–320.

[2] P. Cardullo and R. Kitchin, "Being a 'citizen' in the smart city: up and down the scaffold of smart citizen participation in dublin, ireland," GeoJournal, Jan 2018. [Online]. Available: https://doi.org/10.1007/s10708-018-9845-8

[3] M. Janssen, Y. Charalabidis, and A. Zuiderwijk, "Benefits, adoption barriers and myths of open data and open government," Information systems management, vol. 29, no. 4, 2012, pp. 258–268.

[4] L. G. Anthopoulos, Understanding Smart Cities: A Tool for Smart Government Or an Industrial Trick?. Springer, 2017, vol. 22.

[5] A. Schofield and G. S. Cooper, "Participation in free and open source communities: An empirical study of community members' perceptions," in IFIP International Conference on Open Source Systems. Springer, 2006, pp. 221–231.

[6] W. Ke and P. Zhang, "Motivations for participating in open source software communities: Roles of psychological needs and altruism," PACIS 2008 Proceedings, 2008, p. 76.

[7] G. Fischer, "Communities of interest: Learning through the interaction of multiple knowledge systems," in Proceedings of the 24th IRIS Conference, vol. 1. Department of Information Science, Bergen, 2001, pp. 1–13.

[8] E. Wenger, "Communities of practice: A brief introduction," 2011. [Online]. Available: https://www.ohr.wisc.edu/cop/articles/communities_practice_intro_wenger.pdf

[9] S. Herranz, D. Díez, P. Díaz, and S. R. Hiltz, "Classifying communities for design," in From research to practice in the design of cooperative systems: Results and open challenges. Springer, 2012, pp. 97–110.

[10] C. Reddick and L. Anthopoulos, "Interactions with e-government, new digital media and traditional channel choices: citizen-initiated factors," Transforming Government: People, Process and Policy, vol. 8, no. 3, 2014, pp. 398–419.