

Provenance in Social Computing: A Case Study

Mirela Riveni ^{#1}, Mohamed Jehad Baeth ^{*2}, Mehmet S. Aktas ^{*3}, Schahram Dustdar ^{#4}

*#Distributed Systems Group, TU Wien
Vienna, Austria*

¹m.riveni@infosys.tuwien.ac.at

⁴dustdar@dsg.tuwien.ac.at

**Computer Engineering Department,
Yildiz Technical University
Istanbul, Turkey*

²mohamed.jehad.baeth@std.yildiz.edu.tr

³aktas@yildiz.edu.tr

Abstract—Complex systems such as Collective Adaptive Systems that include a variety of resources, are increasingly being designed to include people in task-execution, and so social computing is not a stand-alone paradigm only, but it is increasingly researched within mixed-resource systems. The Social Computing paradigm has led to significant advancements in engaging people as resources and/services in solving tasks that can not yet be solved by software. Collectives, encapsulating human resources/services, represent one type of an application of social computing, within which people with different type of skills can be engaged to solve one common problem or work on the same project. Mechanisms of managing social collectives are dependent on functional and non-functional parameters of members of social collectives. In this work, we investigate and show experimental results of how provenance data related to those parameters can help better visualize and extract interaction and performance patterns during a collective's run-time.

I. INTRODUCTION

Large-scale distributed systems, such as Collective Adaptive Systems (CAS) are increasingly designed to include various types of resources and stakeholders, such as Cloud resources and services, Internet of Things (IoT) devices and services, and in the last decade *people*, who are included as resources and services for tasks that cannot yet be solved and executed by software only. Thus, today we talk and work about mixed systems that include hardware, software and people (e.g., [1], [2], [3], [4]). This work is focused on the part of these types of systems that concerns the management of human-based resources/services.

In line with the previous work of some of the authors of this paper, we use the concept of Social Compute Units (SCU) to refer to a collective of people, working for a common goal, and who are socially connected in the context of that collective. SCUs are collectives that can be included in socio-technical systems for tasks intended for humans, and managed in an automated way. We name individuals belonging to an SCU as Individual Compute Units (ICUs). In this paper we use the terms worker and ICU interchangeably. SCUs are introduced in [5]. Examples where collectives such as SCUs are used for task execution include: incident management in IT service management [6], incident management in facility management, software-development collectives, online language translations

[7] etc. With the SCU concept, the collectives in all of the mentioned examples are thought to be automatically or semi-automatically managed. In addition, SCUs are thought to be managed elastically, similarly to the Cloud paradigm, where workers can be added and/or excluded from an SCU at run-time; they can be replaced if not performing as required or they can be excluded if the capabilities that they possess are not needed at particular points, new ICUs can be added to the collective if new types of tasks are generated with requirements that the members of the collective do not possess. Thus, SCUs are elastic in multiple dimensions, such as the number of members, topology, as well as included member-capabilities and performance level.

While appreciable work exists on metrics, monitoring strategies and adaptations for collectives such as SCUs, little work exists on utilizing provenance data in social computing. However, we believe that provenance data can help social computing systems in several contexts, such as: a) extracting different behavior patterns for workers, based on which management mechanisms can be improved; b) easier visualization of events for business stakeholders; c) sharing of a common provenance model by multiple social computing environments, which would give a better overview of the capabilities of workers, and most importantly it will allow interoperability between different worker pool platforms so that workers from several different provisioning platforms can be invoked to work in a common collective.

The main contribution of this paper is to investigate how provenance data can be used in social computing and the benefits it can offer for task and worker management, taking the concept of SCUs as our case study.¹ A provenance data model named PROV-DM [8] has been standardized by W3C on which we base our discussion throughout this work. Other models and extensions exist as well, e.g., [9].

The rest of this paper is organized as follows: Section II presents our motivation and research questions derived from a discussion on social-computing provisioning and management

¹We do not use the term social computing here to refer to social networks but to human-based task execution in complex systems such as CAS.

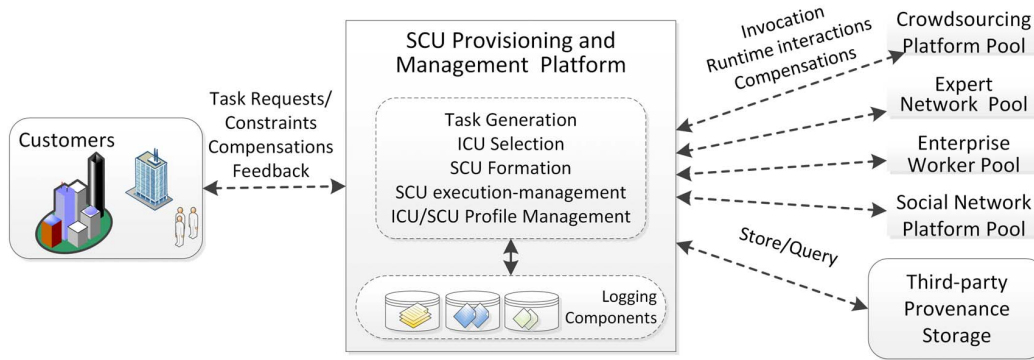


Fig. 1: SCU environment model.

mechanisms in which provenance data can be utilized. In section III we discuss an SCU platform and how provenance-provisioning can fit within the model. In section IV we present our experiments, providing a proof-of-concept implementation of an SCU execution, including logging provenance data and visualizing provenance-data for parts of an SCU lifecycle. Section V discusses related work, while we conclude the paper in Section VI.

II. MOTIVATION AND RESEARCH QUESTIONS

A. Provenance data in social-computing management-mechanisms

We describe next, the most important cases where performance and interaction data is crucial for social computing systems, both from the perspective of an individual and social collectives, such as the aforementioned concept of SCUs.

1) *Individual Task-assignment and Formation of collectives/Social Compute Units*: Whether it is for individual task execution or for task assignment in collectives, workers are ranked based on several pre-set metrics regarding their performance, with which the appropriateness for a task is assessed. Worker selection algorithms are also run during runtime, when a new worker is needed due to different events, such as unexpected task generations, or insufficient capacity of a collective to handle tasks.

2) *Adaptation mechanisms for Social Compute Units*: Defining, modeling and measuring workers' performance is also important for novel systems that enable elastic adaptation of Social Compute Units. Algorithms that calculate ICU performance can be used with adaptation mechanisms that include task delegations based on events. A task can be delegated at runtime, to a worker who had not been included in the collective from start. Consequently, with elastic-adaptations of collectives, a worker can be added to a collective as well as excluded from a collective at run-time [10]. After each adaptation, and at the end of each SCU execution, worker metrics are updated.

3) *Misbehavior prevention and False negatives in Misbehavior detection*: Worker misbehavior, such as assigning a

lower vote to another worker on purpose, tricking the system by sometimes uploading non-complete or unsatisfactory results and still getting rewarded and other times uploading satisfactory results are common in human computation in general. With available provenance data of workers over a period of time, misbehavior detection mechanisms can be developed, by extracting different misbehavior patterns and models from that data. Moreover, mistakes can be treated as a misbehavior sometimes, and if there is a way to visualize which worker did what during an SCU execution, mistakes can be found more easily and existing misbehavior patterns corrected.

4) *Incentive Mechanisms*: Social computing systems employ monetary or non-monetary based incentive mechanisms. Some incentive mechanisms are based on updating *trust* and *reputation* metrics for workers, which are defined considering performance-based and interaction-based historical data of workers, whereas some incentive mechanisms are purely monetary. Incentive models utilize worker interests as well as behavior over time beginning from the starting point of worker engagement in a system, and in this way different incentive plans for each worker can be built.

5) *Compensations*: Closely related to incentives, compensation mechanisms are also based on monetary and/or non-monetary rewards or sanctions, both for workers and customers. One of the most common compensation types is updating worker and customer reputation. Once again, these mechanisms are based on appropriate metrics with which worker behavior is monitored over time, as well as metrics for customer behavior, e.g., their payment method and behaviors, and/or sincerity when specifying their satisfaction from the received results. Regardless of the motivation of people, whether they want to work voluntarily to benefit a cause, or to get paid, workers need to have a clear overview of their progress and of the way in which their work was rewarded or sanctioned over time, so that they can make better decisions as how to act in their future engagements in social computing systems.

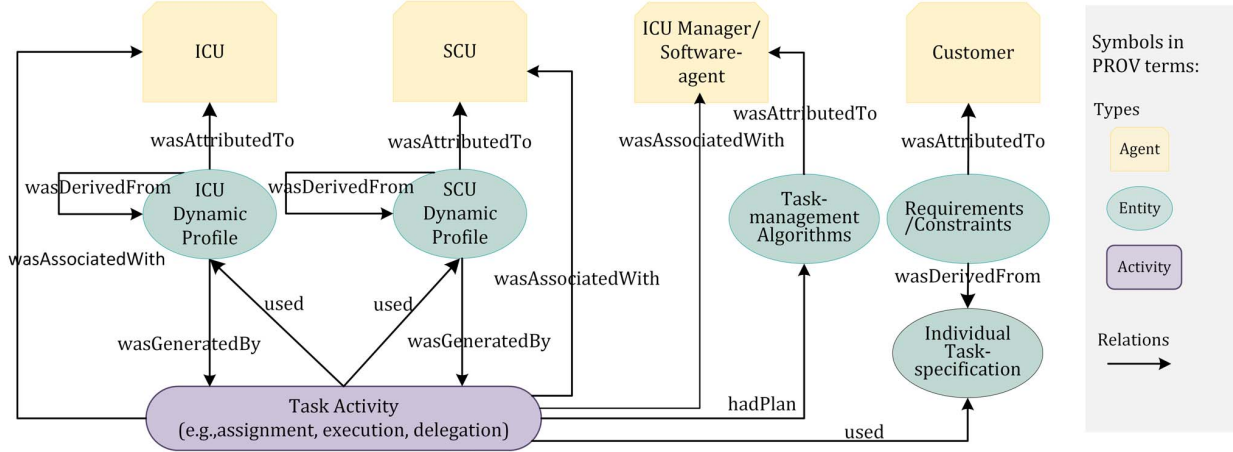


Fig. 2: A graphical provenance-model based on the PROV-O specification, focusing on ICU task executions and profile updates.

B. Challenges

All of the aforementioned strategies, require data regarding workers, and most of it depends on data that is updated based on workers' history of interactions and performance. Thus, provenance can be a good fit to represent historical data for workers. We hypothesize that with provenance-data, the aforementioned mechanisms would serve social computing provisioning-systems in making them more efficient by focusing on providing qualitative work to get high-quality results, in shorter time and on a lower budget, than expected from cases when provenance data is not utilized. Several research questions arise from our hypothesis:

- Can provenance data help us in conceiving, modeling and defining novel metrics that could be used in worker/ICU ranking and selection algorithms for collective formations as well as for elastic-adaptation mechanisms? Consequently, can we enhance existing trust and reputation models, incentive mechanisms and rewarding with new metrics derived from provenance data?
- Can we predict which ICUs collaborate well with which ICUs based on provenance-data of their performance and interactions within the same SCU?
- How can a worker trace his/her progress across several SCUs and extract relevant information that would benefit his/her future behavior in being more productive, and contribute more efficiently in future engagements?
- Can we trace the events that brought to SCU adaptations and the parameters of the SCU adaptation due to those events, so that in future executions of SCUs with similar customer requirements and constraints, adaptations could be more efficient due to action plans that can be devised from provenance data that trace SCU executions?

In this paper we approach the first challenge and investigate the extraction of data that can be used in defining ICU parameters/metrics.

III. SCU ENVIRONMENT AND PROVENANCE

Fig. 1 shows a model of an SCU provisioning platform. We don't show specific components as this is not the focus of this work, but some core management mechanisms of an SCU platform are listed. SCUs have their own lifecycle, so customers that can be companies, different organizations, private individuals etc submit their task request to the SCU provisioning platform, the platform then generates tasks based on the requirements and constraints given, e.g., required skills, cost, deadline. An ICU selection module, which is a task-to-ICU matching module returns the most appropriate ICU for each task. A collective formation module is a composition module, and an orchestrator. ICUs and SCUs have their own profile which is monitored and updated at runtime and at the end of the SCU execution. Customers can give feedback about the final results, and ICUs are rewarded or sanctioned at the end of the SCU execution. Depending on the domain, an SCU provisioning platform may have its proprietary pool of resources, but human-based resources/services can also be invoked from multiple external sources.

From the perspective of a specific SCU environment, a provenance approach to logging can give an overview of historical data useful for the mechanisms supporting social collective provisioning and management. On the other hand, if we assume that there can be multiple social computing milieus and that a third-party can provide provenance storage for multiple SCU provisioning platforms, workers can be treated equally regardless of the fact on which platform they have been registered, and from which SCU environment they are invoked. This brings to interoperability of multiple SCU environments, which will allow for a common large-scale pool of human-based resources bringing flexibility to the types of social-computing applications.

A. Privacy considerations

Although not specifically related to our focus in this work, we want to make notice of privacy issues regarding provenance data, because provenance data can be a double-edge sword.

Current social computing systems in the industry all base their operations of sensitive personal data, for verification of worker and customer profiles. Some authors argue that provenance can help reduce privacy issues by providing data that help in authorization and access control mechanisms [11]. However, this approach tackles access issues and protection against access risks, while true privacy means that if a worker and/or a customer wishes not to be identified it is his basic human right not to be, and this should be reflected in system design. Hence, we advocate that workers in a system be identified by Ids if so desired, and provenance data to be captured only related to an id, and only regarding the performance and interactions between workers. Other authors have also argued provenance to be a major issue for privacy in the context of produced-data and not only of data connected to people profiles, e.g., such as authors in [12]. Last but not the least, payment methods for rewards present a problem, as even if no other personal data is collected, payment methods can be used for identification. We list this issue as an important one to approach when designing provenance-based social computing systems which has to be yet addressed.

IV. EXPERIMENTS

A. Setup

We have implemented a Java-based prototype that is a simulated social-computing environment. We have modeled an ICU with static properties such as Id and skill type, and dynamic properties, some of which are descriptive e.g. cost per task, and others that are mathematically defined and calculable. Thus, we designed metrics that we have identified as relevant for ICUs, and that serve as key performance indicators for them as dynamic properties. The most important metrics to mention here are: *effort*, *productivity*, *willingness*, *reliability*, and *performance trust*, which is a weighted aggregate metric of the aforementioned ones; *social trust*, which we define as a weighted aggregate of votes from people that the worker has collaborated with, and *socio-technical trust* which is a weighted average mean of the aforementioned performance-based trust and social-trust metrics. For the definitions of the aforementioned metrics, we refer you to a previous work of some of the authors of this work, presented in [13]. We modeled SCUs as lists of ICUs, which in turn have their own metrics. A task is designed with specific properties, such as the skill type required to execute it, the cost for executing it, and the deadline to execute it.

B. Experiment type and Result analysis

1) *Dataset*: To test what type of provenance data can we store and how the visualization of this data will help us interpret results in a social computing environment in an efficient way, we implemented a specific task-execution and worker/ICU management mechanism. We generated 200 ICUs with different skill types and costs per task, as well as random initial values for the aforementioned metrics, all of them in the (0,1] interval. For a faster algorithm run, instead of ranking ICUs for each task assignment, we ran a ranking algorithm

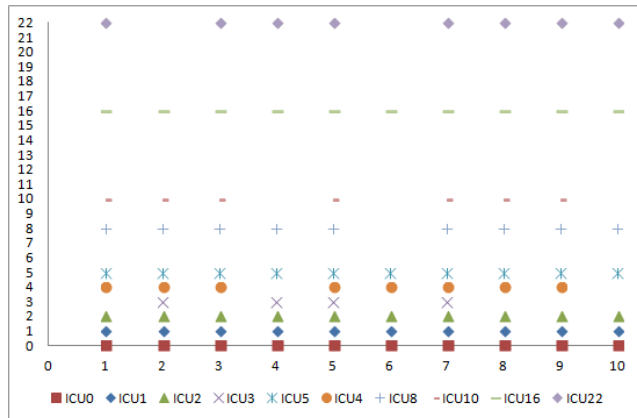


Fig. 3: ICUs across 10 checkpoints of an SCU run.

first, to rank all ICUs, regardless of their skill type, based on the weighted values of three different metrics, given as an input: social trust, reputation and reliability. For ranking we implemented an algorithm based on the Analytic Hierarchy Process model (the description of which is out of the scope of this paper).

Next, we ran a task assignment and scheduling algorithm as in the following: we generated a bag of tasks with 40-50 tasks and assigned them in a FIFO order to ICUs from the ranked list, this time matching the skills of workers with those of the required ones for each task. The ICUs selected in the initial assignment form an SCU. We ran sequentially 10 bags of tasks, each time assigning 40-50 tasks to ICUs, after each execution of a bag-of-tasks. Our scheduling algorithm was designed such that it allowed for elastic adaptation of the initial SCU, such that tasks that reached a threshold in the queue of an ICU were delegated to other ICUs. Delegated tasks were assigned either to ICUs already within the SCU or to an ICU from the pool of ICUs from the ranked list, depending on a willingness value of ICUs for executing a task and availability. The willingness value was set randomly to 0 (not willing) or 1 (willing). ICUs that did not have any task assigned in a run, were excluded from the SCU, and new ones to whom task were assigned were added to the SCU. Hence the number of ICUs in each run fluctuated, and with this we updated the metrics, indicators of their performance and interactions, after each bag of tasks that was assigned and executed by the collective/SCU. For clarity, we denote the update of the properties of SCU members after each bag-of-task assignment, as a checkpoint. Thus, a checkpoint points to one bag-of-tasks execution. We generated a log file of the SCU execution, using the Apache log4j logging utility, storing metric values for every SCU member for each checkpoint.

2) *Provenance*: Utilizing the log4j log file, we mapped ICUs, SCUs, Tasks, and events during the SCU execution (e.g., task assignment, delegation and ICU profile updates) to provenance notation. We conducted the mapping using ProvToolbox [14], which allows for creating PROV documents in Java. Thus, with ProvToolbox, we generated XML files

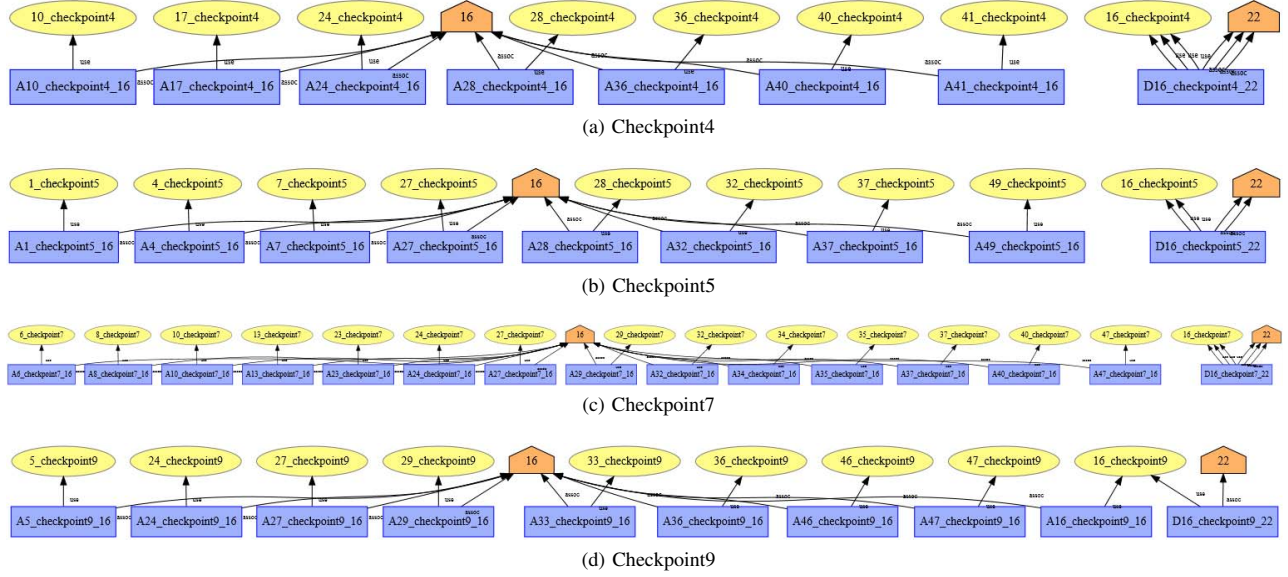


Fig. 4: Tasks for ICU 16 and 22 at four selected checkpoints during one SCU execution.

with provenance tags, as well as provenance graphs that reflect provenance types and relations from the mapping. Our PROV-DM-based [8] model on which we based the mapping is shown on Fig.2. ICUs, SCUs, Customers, and the Task Manager and ICU Manager represent *Agents*. ICU and SCU profiles in which we store data for each ICU and the SCU respectively, as well as task-related requirements and constrains represent *Entities*. In addition, task requirements are also *Entities* (not shown on the figure for simplicity purposes). *Activities* in our implementation are the following actions: task *assignment*, task *delegation*, successful task *execution*, and ICU *profile update* after each execution.

Let us examine some results and some information that can be extracted from looking at the visualization graphs of provenance data that we extracted from our log file. We chose to analyze two ICUs from the provenance logs, the ICU with Id 16, and the one with Id 22. We can see that tasks were delegated from ICU 16 to ICU 22 during multiple time points of the SCU execution, as follows: two tasks in the first and the fifth checkpoint (Fig. 4b); three in the third checkpoint, the fourth (Fig. 4a), and the seventh (Fig. 4c); one task in the eighth, ninth (Fig. 4d) and tenth checkpoint; no delegated tasks in the second and sixth bag-of-tasks assignment. Due to the provenance visualization, this data can be easily inferred without queries, which could be useful for business stakeholders.

Fig. 4 presents the visualization of executed assigned tasks as well as delegated and executed tasks for ICUs with Id 16 and 22, for a few checkpoints. Only from looking at these two ICUs we identified two possible deductions that can be defined as metrics. We describe them in the following subsection.

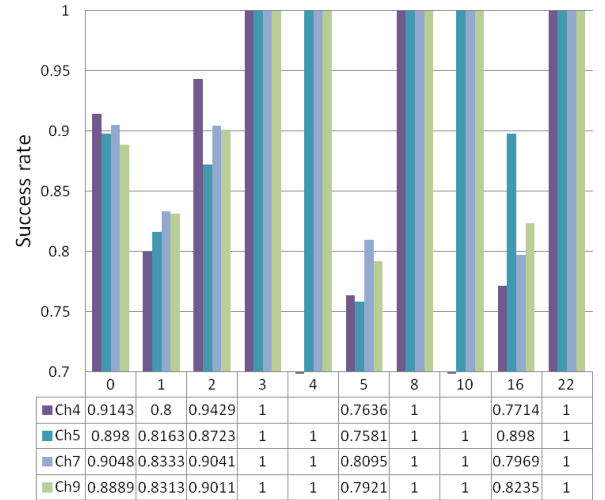


Fig. 5: ICUs and tasks during the execution of one bag-of-tasks. The graph result of checkpoint 7.

C. Metrics based on extracted information from provenance data

1) *Delegation-based Profile similarity*: The graphs in Fig. 4 show assignments and delegations for ICU 16 and ICU 22 in four separate checkpoints. Assignment activities are denoted with A and delegations are denoted with D. Agents are ICUs (in orange color) shown only with their Ids, while entities on the graphs are task descriptions, where we show only their Ids and the specific checkpoints in which they were assigned (in yellow oval curves). Activities show the Id of tasks and associated ICUs. From the experiment results we can deduct that ICUs with Id 16 and Id 22 have the same skill type,

because tasks from ICU Id 16 were only delegated to the ICU with Id 22. This also testifies for the consistency of ICU 22 in the sense that ICU 22 was only invoked in the SCU when some tasks from ICU 16 needed to be delegated. It means that ICU 22 was successful in executing the tasks delegated to it, and thus was invoked multiple times. Fig. 5 shows the success rate of all ICUs at four checkpoints, which also proves our deduction from the provenance graphs, that ICU 22 executed all delegated tasks from ICU 16, as ICU 16 has a success rate that does not achieve a value of 1 (a value of 1 indicates that all assigned tasks were successfully executed), while ICU 22 has a success rate of 1 at each checkpoint.

Every time tasks needed to be delegated from ICU 16, ICU 22 was included in the SCU, this means that ICU 16 and ICU 22 do not provide the same skill type only, but also that they are close with their reputation scores (because we ranked ICUs based on their reputation scores initially). Hence, provenance-data regarding delegations can be a good indicator for the fact that delegations can be used in metrics that define profile similarity between two ICUs.

Let us elaborate more on this. Let us denote, three ICUs v , u and w , which when mapped to our model and results, v denotes ICU 16 which is included in the collective from start, while u represents ICU 22 and w is another ICU with the same skill as ICU 22. Let us define $d(v,u)$ denoting the interaction intensity between v and u in the $[0,1]$ range, defined by the number of delegation relations between v and u across multiple checkpoints. If we denote the availability of ICUs with a_v , a_u and a_w with 0 if not available, and 1 if an ICU is available, and the reputation of all three ICUs with r_v , r_u , r_w , then we can consider the following relations to be valid at one single checkpoint: a) if a_u a_w are both 1, and the value of $d(v,u)$ is within the $(0,1]$ range, while the value of $d(v,w)$ is 0, then we can safely assume that the reputation relation of v , u , w to be $r_v > r_u > r_w$, and b) if $a_u = 0$ and $a_w = 1$, and $d(v,u)$ has some value from the $(0,1]$ range, while the value of $d(v,w)$ is 1, then we can safely assume that the reputation score of u and w are close $r_u \approx r_w$, in addition of the validity of the $r_v > r_u > r_w$ relation, particularly when the worker pool is large. These conclusions are intuitive if one knows how ICUs are ranked, because w is ranked as the next appropriate worker after u . However, for analysts who do not know the details behind the mechanisms of ICU selection, it can be a valuable information. Moreover, this information is even more valuable when the worker pool with which an SCU environment works is comprised of workers registered on multiple platforms, and if analytics is conducted on Big-data.

Concretely, from our algorithm, we came to formulate a similarity metric based on reputation and skill. As aforementioned, for each task, we have a ranked list of reserve resources/workers with the appropriate skill, and they are ranked by their reputation values in a descending order. So, we define a similarity metrics based on reputation and skill-type, as in the following:

$$s_{vu} = \frac{rank_u}{m} * \frac{d_{vu}}{t_v}, \quad (1)$$

where d_{vu} denotes the number of delegated tasks from v to u , number of total assigned tasks to v , and m is the total number of workers in the reserve list. The rank value of the workers, starts with the value that represents the total number of workers in the list and continues in a descending manner. The similarity metric can have values in the range $(0,1]$, where a higher value means more similar profiles (similar reputation scores). So, for example if v is the first ranked resources in a ranked list of 100 resources and u is the third, and v delegated 6 tasks out of 10 to u , then the similarity value between v and u will be:

$$s_{vu} = \frac{98}{100} * \frac{6}{10} = 0.588.$$

Hence, data-provenance regarding delegations can help in defining novel metrics for ICUs, and we demonstrated this by arguing that profile similarity with regards to two parameters, ie. skill-type and reputation, can be defined by having a detailed overview of task-delegations.

2) *Skill utility for the SCU*: By looking at the provenance-graphs we can conclude that ICUs with Id, 0,1,2,16 posses the core skills which are crucial for the SCU, as these ICUs were continuously present in the SCU during its whole lifecycle. However, we can also infer that the possession of the core skills needed for the SCU is also true for ICU 22, based on the fact that tasks were delegated only from ICU with Id 16 to it and not another ICU. As an illustration, if we consider a software-development scenario, from the graphs it is possible to conclude that the skills of ICUs that are continuously included in the SCU are programming skills, and other ICUs posses other skills such as for example design skills, which need not be utilized during the whole development phase. For comparison, Fig. 3 shows which ICUs were included at which time-point within the SCUs runtime, where the x-axis denotes the ten checkpoints of the SCU run. Fig. 6 shows a provenance visualization of all ICUs, with assignments and delegations at checkpoint 7. The graphs also show which specific tasks were delegated to which ICUs, e.g., tasks with Ids 32, 35 and 37 were delegated from ICU 16 to ICU 22.

The results of our experiment show that provenance data can be successfully applied to social computing environments and the visualization of provenance data can help in deriving interaction and performance related information about workers in an SCU environment. This, can in turn help in deriving novel metrics for ICUs. We discussed two deductions from the visualization of the experiment results in provenance graphs, providing a metric based on the appropriateness of ICUs for specific tasks, in terms of skill and trust, and executed delegated tasks.

V. RELATED WORK

Markovic et al. in [15] pose several research questions related the utilization of provenance in social computing, mainly focusing on the assessment of worker trust, and selection of

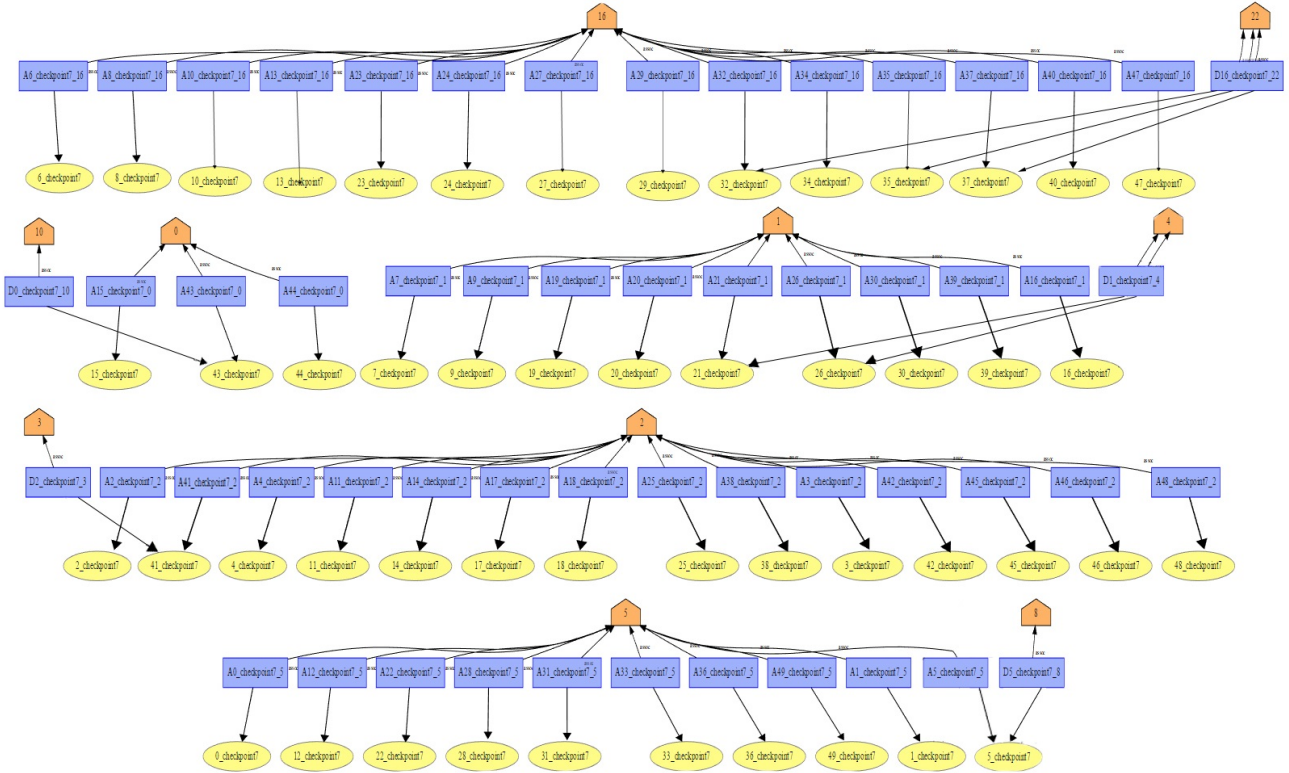


Fig. 6: ICUs and tasks during the execution of one bag-of-tasks. The graph result of checkpoint 7.

appropriate workers for tasks. In addition, by discussing a concrete scenario of social computing, they also argue that some concepts, such as task delegations and incentives are still problematic to express using Prov-DM [8]. A follow up work by the same authors presented in [16] describe an extension to PROV-O [17] and P-PLAN intended to benefit social computing scenarios. Data-provenance modeling for group-centric collaborations is presented in [18].

Packer et al., discuss provenance for CAS in [19] and argue that provenance data helps in making CAS more transparent and accountable, and also help in assessment of users' trust. The authors argue that provenance makes systems more accountable as provenance data provides information about the use of data and decision-making processes. The former, meaning the way user data is used by the system is also connected to systems' transparency, because users could be provided with a timely and detailed view of which type of data provided by them was used by the system and how. Moreover, provenance data may be utilized for enforcing privacy rules. The authors provide several use cases where provenance can be beneficial, including cases where it may give enough information so that the system users can be informed about the way their reputation is calculated, making a similar justification for provenance data, to one of our justifications in this work. Authors in [20] have investigated whether existing provenance systems are capable of handling large-size social-provenance

data and provided their own decentralized architecture model that could better handle provenance-data as opposed to existing ones, in terms of scalability, data quality and privacy concerns. The work presented by authors in [21] discusses a large-scale synthetic social provenance database, designed for social networks, and how that provenance data can be used to define and calculate metrics such as credibility of a person. These type of databases can be useful to motivate research of large scale databases for more complex social-computing systems,

A case where provenance data can be used in a specific domain, such as in crowdsourced data analysis tasks is discussed in [22]. Another domain-based example regarding provenance and related to geospatial data is given in [23], where the authors mention processes with human-in-the-loop. Needless to say, we mention the two latter works as two of several existing works, which testify that provenance can also be used in providing a better Quality of Result, in addition to enabling more efficient management of task executions and resources/workers.

VI. CONCLUSION

In this paper, we have discussed the most important mechanisms in social computing that would benefit from provenance data. We presented an implementation of an SCU-provisioning and management strategy, the execution trace of which we mapped to provenance notation based on the PROV-DM model. Our experiment showed that provenance data can help

in deriving information that helps in building better worker profiles as well as more efficient management mechanisms due to metrics that can be extracted from their provenance data. In future work, we plan to work on provenance in mixed systems, and investigate how management mechanisms of mixed-resources can benefit from provenance. One concrete challenge is to investigate whether provenance-based data can help us detect if a resource/service is a software piece or a human-based service.

REFERENCES

- [1] S. Dustdar and H. L. Truong, "Virtualizing software and humans for elastic processes in multiple clouds- a service management perspective," *IJNGC*, vol. 3, no. 2, 2012.
- [2] H.-L. Truong and S. Dustdar, *Context-Aware Programming for Hybrid and Diversity-Aware Collective Adaptive Systems*. Cham: Springer International Publishing, 2015, pp. 145–157. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-15895-2_13
- [3] D. Miornadi and L. Maggi, "Programming social collective intelligence," in *Technology and Society Magazine, IEEE*. IEEE, 2014, vol. 33, pp. 55–61.
- [4] H. Zhuge, "The future interconnection environment," *Computer*, vol. 38, pp. 27–33, 2005.
- [5] S. Dustdar and K. Bhattacharya, "The social compute unit," *IEEE Internet Computing*, vol. 15, pp. 64–69, May 2011. [Online]. Available: <http://dx.doi.org/10.1109/MIC.2011.68>
- [6] B. Sengupta, A. Jain, K. Bhattacharya, H. L. Truong, and S. Dustdar, "Who do you call? problem resolution through social compute units," in *ICSOC*, ser. Lecture Notes in Computer Science, C. Liu, H. Ludwig, F. Toumani, and Q. Yu, Eds., vol. 7636. Springer, 2012, pp. 48–62. [Online]. Available: <http://dblp.uni-trier.de/db/conf/icsoc/icsoc2012.html#SenguptaJBTD12>
- [7] V. Ambati, S. Vogel, and J. Carbonell, "Collaborative workflow for crowdsourcing translation," in *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, ser. CSCW '12. New York, NY, USA: ACM, 2012, pp. 1191–1194. [Online]. Available: <http://doi.acm.org/10.1145/2145204.2145382>
- [8] P. Missier and L. Moreau, "PROV-dm: The PROV data model," W3C, W3C Recommendation, Apr. 2013, <http://www.w3.org/TR/2013/REC-prov-dm-20130430/>.
- [9] L. Moreau, D. Li, F. Joe, G. V. Daniel, G. Paul, J. Mike, M. Simon, M. Paolo, P. Jeff, and Z. Jun, "Open provenance model (opm) owl specification," Tech. Rep., 2010, <http://openprovenance.org/model/opmo>.
- [10] M. Riveni, H. L. Truong, and S. Dustdar, "On the elasticity of social compute units," in *Advanced Information Systems Engineering - 26th International Conference, CAiSE 2014, Thessaloniki, Greece, June 16-20, 2014. Proceedings*, 2014, pp. 364–378. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-07881-6_25
- [11] Y. Cheng, D. Nguyen, K. Bijon, R. Krishnan, J. Park, and R. Sandhu, "Towards provenance and risk-awareness in social computing," in *Proceedings of the First International Workshop on Secure and Resilient Architectures and Systems*, ser. SRAS '12. New York, NY, USA: ACM, 2012, pp. 25–30. [Online]. Available: <http://doi.acm.org/10.1145/2420936.2420941>
- [12] S. B. Davidson, S. Khanna, S. Roy, J. Stoyanovich, V. Tannen, and Y. Chen, "On provenance and privacy," in *Proceedings of the 14th International Conference on Database Theory*, ser. ICDT '11. New York, NY, USA: ACM, 2011, pp. 3–10. [Online]. Available: <http://doi.acm.org/10.1145/1938551.1938554>
- [13] M. Riveni, H. L. Truong, and S. Dustdar, "Trust-aware elastic social compute units," in *2015 IEEE TrustCom/BigDataSE/ISPA, Helsinki, Finland, August 20-22, 2015, Volume 1*, 2015, pp. 135–142. [Online]. Available: <http://dx.doi.org/10.1109/Trustcom.2015.367>
- [14] L. Moreau, "Provtoolbox. java library to create and convert w3c prov data model representations." <http://lucmoreau.github.io/ProvToolbox/>, last accessed: 22-4-2017.
- [15] M. Markovic, P. Edwards, and D. Corsar, *Utilising Provenance to Enhance Social Computation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 440–447. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-41338-4_29
- [16] —, *SC-PROV: A Provenance Vocabulary for Social Computation*. Cham: Springer International Publishing, 2015, pp. 285–287. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-16462-5_35
- [17] T. Lebo, S. Sahoo, and D. McGuinness, "PROV-O: The PROV ontology," W3C, W3C Recommendation, Apr. 2013, <http://www.w3.org/TR/2013/REC-prov-o-20130430/>.
- [18] J. Park, D. Nguyen, and R. S. Sandhu, "On data provenance in group-centric secure collaboration," in *7th International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom 2011, Orlando, FL, USA, 15-18 October, 2011*, 2011, pp. 221–230. [Online]. Available: <https://doi.org/10.4108/icst.collaboratecom.2011.247192>
- [19] H. S. Packer, L. Drăgan, and L. Moreau, *An Auditable Reputation Service for Collective Adaptive Systems*. Cham: Springer International Publishing, 2014, pp. 159–184. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-08681-1_8
- [20] Y. Tas, M. J. Baeth, and M. S. Aktas, "An approach to standalone provenance systems for big social provenance data," in *12th International Conference on Semantics, Knowledge and Grids, SKG 2016, Beijing, China, August 15-17, 2016*, 2016, pp. 9–16. [Online]. Available: <https://doi.org/10.1109/SKG.2016.010>
- [21] M. J. Baeth and M. S. Aktas, "A large scale synthetic social provenance database," in *Proceedings of the Ninth International Conference on Advances in Databases, Knowledge, and Data Applications*, ser. DBKDA 2017. 2017, pp. 16–22.
- [22] W. Willett, S. Ginosar, A. Steinitz, B. Hartmann, and M. Agrawala, "Identifying redundancy and exposing provenance in crowdsourced data analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2198–2206, Dec. 2013. [Online]. Available: <http://dx.doi.org/10.1109/TVCG.2013.164>
- [23] D. Garijo, Y. Gil, and A. Harth, "Challenges for provenance analytics over geospatial data," in *Provenance and Annotation of Data and Processes - 5th International Provenance and Annotation Workshop, IPAW 2014, Cologne, Germany, June 9-13, 2014. Revised Selected Papers*, 2014, pp. 261–263. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-16462-5_28