Provenance of Decisions in Emergency Response Environments

Iman Naja, Luc Moreau, Alex Rogers

School of Electronics and Computer Science, University of Southampton
Southampton, SO17 1BJ, UK
{izn1g08,L.Moreau,acr}@ecs.soton.ac.uk

Abstract. Mitigating the devastating ramifications of major disasters requires emergency workers to respond in a maximally efficient way. Information systems can improve their efficiency by organizing their efforts and automating many of their decisions. However, absence of documenting how decisions were made by the system prevents decisions from being reviewed to check the reasons for their making or their compliance with policies. We apply the concept of provenance to decision making in emergency response situations and use the Open Provenance Model to express provenance produced in RoboCup Rescue Simulation. We produce provenance DAGs using a novel OPM profile that conceptualizes decisions in the context of emergency response. Finally, we traverse the OPM DAGs to answer some provenance questions about those decisions.

1 Introduction

Major disasters, like the 2004 Indian Ocean and the 2010 Haiti earthquakes cause deaths, injuries, and serious damage. To minimize the effect of such disasters, emergency responders must work in a maximally efficient way. They must make numerous decisions centered on prioritizing which civilians to rescue and they must make these decisions in unpredictable changing environments while racing against time and coordinating with different rescue agencies. As such, there is an increasing need to build information systems that organize the efforts of responders and improve their efficiency by automating many of the decisions they make on the ground. Most notably, recent efforts in research in the disaster management domain on the levels of developing infrastructure simulation and intelligent agent are being tested in the RoboCup Rescue Simulation league [1].

However, a critical shortcoming arises within current approaches through their inability to represent the causal factors that led certain decisions to be made. In turn, this makes it difficult to determine whether these decisions were compliant with policies and regulations, and to hold decision makers to account[1].

[1] We consider accountability of decisions to be analogous to Weitzner *et al.*'s [25] definition of information accountability, where the transparency of use of information enables ascertaining its appropriate use as per given rules. So, we perceive that transparency of actions and decisions, and how they influenced later actions and decisions, permits the checking of compliance with requirements or policies.

As such, there is a need to document how decisions were made and to refer to such documentation when the need arises to review the history of their making or to check compliance with rules and policies. This can done through recording and querying their provenance, where provenance describes the history of items, physical or immaterial, and how they came to be. Provenance has proven to be useful in a variety of domains including amongst others workflow re-enactment, inferring reasons for result differences in scientific experiments, and quality assurance of data [2,22,16]. To this end, we consider the provenance of decisions to include any data that affected their making as well as the processes that led to this data. We consider this provenance to be vital to understanding causality of events within a system and how decisions influence others in decision chains.

Thus, our work aims to exploit the provenance of decisions to understand how they were made and why. Being motivated by addressing problems in the emergency response domain, we proposed the use case "Provenance of Decision Making in Emergency Response" to the W3C Provenance Incubator Group. The work presented in this paper forms the first steps towards addressing the scenarios of the use case. Accordingly, we use RoboCup Rescue Simulation (RCRS) as a testbed to show how we can enable an automated decision-making system to record its provenance by applying the PrIMe methodology [14] to it. PrIMe assists in indicating what needs to be recorded so that the provenance questions, we are interested in, can be answered. Because answering the questions requires querying provenance graphs, we use the Open Provenance Model (OPM) [17] to produce provenance DAGs, making use of a novel OPM profile that specializes OPM and conceptualizes decisions in the context of emergency response.

In summary, the contributions of this paper are as follows:

1. Applying OPM to the decision making domain, a field in which it has not previously been used. We do so by proposing an OPM profile that specializes OPM and use it to represent decisions in the context of emergency response.
2. A prototype to be integrated with RCRS that generates OPM DAGs and answers provenance queries so as to handle the use case.

The rest of this paper is organized as follows. Section 2 presents our motivation and the use case. Section 3 briefly describes RCRS. Section 4 shows how PrIMe can be used to make RCRS provenance-aware. Section 5 details how provenance information of RCRS can be exposed using the OPM profile *RobocupProfile*. Section 6 presents related work and Section 7 presents future work and concludes.

2 Provenance of Decisions: Tracing Decisions Made by Emergency Responders

We are motivated by the need to interpret events in cases of floods where the police and fire brigade must evacuate casualties according to some prioritization scheme from buildings that are flooded or buildings under the threat of being flooded. Evacuees needing medical attention are taken to a triage area and examined by medics who prioritize their care and delivery to hospitals.
Consequently, we proposed the use case "Provenance of Decision Making: Trac-

ing Decisions Made in Emergency Response Situations"[2] to the W3C Provenance Incubator Group[3] so as to address the need for information systems that not only organize efforts of emergency responders and improve their efficiency but also use the provenance of decisions to reveal how they were made and why. The goal of the use case is to suggest the use of provenance in the Justification for Decisions dimension[4]. This dimension is divided into three sub-dimensions [24]. Currently, we focus only on two: argumentation, where provenance is used to deduce what information affected the choice of a certain solution, and answering why-not questions, where provenance of decisions is used to capture why particular choices were *not* made.

3 Decisions in RoboCup Rescue Simulation

RCRS league is a competition aiming to stimulate research in multi-agent systems in the disaster management domain by inviting participants to devise state-of-the-art strategies that automate decision-making, prioritization, and coordination and cooperation [1]. The simulation models a city hit by an earthquake with fires erupting in various parts of the city and buildings collapsing blocking roads and trapping civilians. Three types of emergency response agents are initially spread across the city with only the knowledge of its map. They then move around learning about the world and performing their tasks. At each time step in the simulation, each agent 'thinks' about what it should do and submits an action to the simulator[5]. This computes the effect of all the agents' actions on the world's state and informs the agent about the effect of its action and what new entities or changes it should sense.
Due to space restrictions, we focus only on ambulance teams that remove civilians trapped in buildings and transport them to refuges[6].

Ambulance Agents We utilize the platform's default ambulance agents after slightly improving them so they behave as follows. Each agent prioritizes civilians according to how far they are from it, irrespective of criticality of conditions of other civilians. So, it sorts the civilians, plans its path to the nearest one, heads to it, unburies it, loads it, plans its path to the closest refuge and moves there. Once at the refuge it unloads the civilian and repeats the previous steps for its next target. If it is not aware of any civilians, it wanders about until it finds one. Based on this scheme, when an agent heading to a civilian discovers another on its way, it re-prioritizes and chooses the closer one. Also, an agent informs other agents when it rescues a civilian or discovers that one has perished. This prevents cases where agents head to save civilians that have died or have already been rescued and cases where agents wander about looking for civilians while others are aware of ones that need to be rescued.

[2] `http://tiny.cc/Prov_Decision_Making`
[3] `www.w3.org/2005/Incubator/prov/wiki/Main_Page`
[4] The use case includes additional propositions, see Future Work (§7).
[5] The simulator is composed of a mediator kernel and several specialized simulators.
[6] The other two types of agents are Fire Brigades which extinguish fires in buildings to prevent further damage to them and Police forces which clear blockages in roads.

4 Provenance-Aware RoboCup Rescue Simulation

We now show how to apply PrIME [14], a three-phase methodology that when applied to a system makes it provenance-aware, to enable RCRS to record its provenance. In the first phase of PrIMe, we identified the following questions as relevant to understanding events that usually take place in RCRS:

1. A civilian C1 was rescued by agent A. What were the steps (i.e. the sequence of actions) that A took to rescue C1?
2. A certain civilian C2 died. Why was C2 not rescued?
3. After A rescued C1, its prioritized target list had C2 on top. However, A rescued C3 next. What pieces of information influenced that change of goals?
4. What were the factors that led to the long delay in saving C4?

Note that PrIMe supports adding anticipated use cases at later stages. In the second phase of PrIMe, we decomposed RCRS into three actors representing the ambulance agent, kernel, and simulator. We then iterated the second phase to identify the actors within the ambulance agent that are responsible for the different decisions, and they are as follows. *(1) Thinker*: the component that decides what to do next based on the strategy and the state. It is further decomposed into two actors: the *State maintainer* maintains the agent's state and view of the world and the *Planner* decides what action to perform next. *(2) Path Searcher*: the component that uses a path search algorithm to plan the path from one place to another. *(3) Sorter*: the component that sorts the list of target civilians based on a prioritization scheme. In the third phase of PrIMe, we mapped actors and messages to OPM processes and artifacts respectively and created OPM edges corresponding to information flow between the actors.

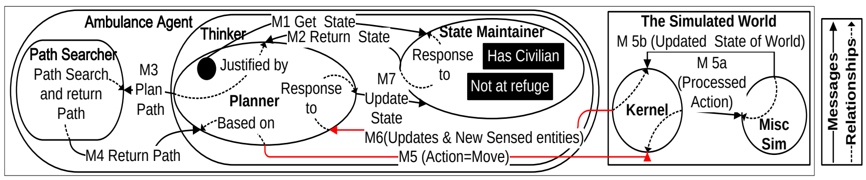
Figure 1 shows an example of a decision reached with the involvement of some RCRS's actors and illustrates the flow of messages between them[7]. We trace the actors' decisions and their interactions and state what interesting process documentation they record. First, *Planner* checks the state of the agent by consulting *State Maintainer (message M1)*. *State Maintainer* asserts that the agent is currently carrying a civilian and that it is not at a refuge *(M2)*. Based on this state, *Planner* decides to move towards a refuge. It requests from *Path Searcher* the shortest path to the closest refuge *(M3)*. *Path Searcher* produces the path, asserts it, and returns it to *Planner (M4)*, which then informs the kernel that it wishes to execute a 'move' action *(M5)*. The kernel processes the action and replies to the agent with the updated state of the world, newly perceived entities, and agents' messages *(M6)*. Finally, *Planner* uses this response to asserts the new information and update *State Maintainer (M7)*.

5 OPM-based RCRS Provenance Information

OPM is a model of provenance designed to, among other requirements, allow the exchange of provenance information between systems [17]. We assume that the reader is familiar with its basic concepts. We chose OPM because of the features it possesses, like controlled vocabulary, annotations, inference rules, and profiles.

[7] For an example involving all RCRS's actors check `http://tiny.cc/iznMoveActors`

Fig. 1: Some RCRS Actors and their Interactions



An OPM profile consists of a mandatory unique global identifier in addition to four optional elements as follows:

1. Controlled vocabulary for annotations, and their permitted subjects and values, specifying application-specific properties. These are used to subtype nodes and edges of OPM DAGs and to define application-specific properties.
2. General guidelines to how OPM graphs can be structured.
3. Profile expansion rules that show how nodes or edges can be derived.
4. Syntactic shortcuts and how they can be serialized.

We now present the OPM profile *RobocupProfile* that specializes OPM to represent provenance produced in RCRS. At this stage we only utilize the first two elements of OPM profiles. First, we specify two subtypes of the Agent node *(1) Ambulance*, corresponds to ambulance agents, and *(2) Kernel*, corresponds to the kernel. Also, accounts are classified into *(1) KernelAcct* - corresponds to the kernel's viewpoint, *(2) AgentAcct* - corresponds to the viewpoint of agents, and *(3) AgentDetailAcct* - corresponds to the nodes and edges pertaining to the internal processings of the agent. Finally, tables 1, and 2 display the controlled vocabulary of *RobocupProfile*.

RobocupProfile explicitly shows how processes and artifacts and the dependencies linking them can model how RCRS ambulance agents make their decisions. Specifically, the dependencies within each decision process on the different artifacts are explicitly stated. In turn, the dependencies of those artifacts on other artifacts are also declared. Further, the dependencies of those artifacts on previous decision process are stated, using subtypes of *was generated by* edges. Hence, chains of decisions, and how their results came out, can be expressed. Figure 2 shows a portion of an OPM DAG illustrating two *AmbulanceActions* (unload and move), the decisions that produced them, and the artifacts that influenced those decisions. In more detail, *TaskResult* 'Civilian C1 rescued by

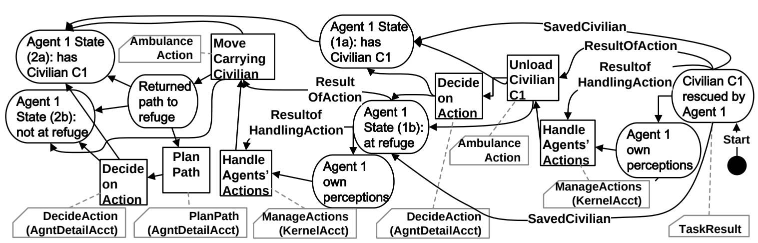
	Sub-type	Account
Artifacts	*TaskResult* (rescuing a civilian succeeded or failed), *AgentPerceptions*, *AgentMessage* *AgentState*, (e.g. agent's target list, position, whether it is carrying a civilian or not)	*AgentAcct* ∪ *KernelAcct*
	SortedListCivs, *PlannedPath*	*AgentDetailAcct*
Processes	*AmbulanceAction* (move, unbury, load civilian, unload civilain, rest), *PassMessages*, *ReceiveMessage*	*AgentAcct* ∪ *KernelAcct*
	DecideAction, *PlanPath*, *SortCivilians*	*AgentDetailAcct*
	ManageMessages, *ManageActions*	*KernelAcct*

Table 1: RobocupProfile Artifacts and Processes and the Accounts they belong to.

Edge	Sub-type	Effect	Cause
Used	*ManagingMssgs*	*ManageMessages*	*AgentMessage*
Used	*ConstructingMssgs*	*PassMessage*	*AgentMessage*
WGB	*SortedListGeneration*	*SortedListCivs*	*SortCivilians*
WGB	*PathGeneration*	*PlannedPath*	*PlanPath*
WGB	*ResultOfAction*	*TaskResult* ∪ *AgentState*	*AmbulanceAction*
WGB	*ResultOfHandlingAction*	*TaskResult* ∪ *AgentState*	*ManageMessages*
WGB	*DecomposingMssgs*	*AgentMessage*	*ReceiveMessages*
WTB	*ActionHandling*	*ManageActions*	*AmbulanceAction*
WTB	*DecidingAction*	*AmbulanceAction*	*DecideAction*
WDF	*TargetDiscovery*	*AgentState*	*AgentMessage* ∪ *AgentPerceptions*
WDF	*SavedCivilian*	*TaskResult*	*AgentState*
WDF	*UpdatedState*	*AgentState*	*AgentState* ∪ *SortedListCivs*
WDF	*SortedCivs*	*AgentState*	*SortedListCivs*
WDF	*PathToPriority*	*PlannedPath*	*AgentState*

Table 2: RobocupProfile Edges

Fig. 2: Portion of OPM DAG



Agent 1' was derived from *AgentPerceptions* which were generated by the kernel managing agents' actions. This, in turn, was triggered by the agent unloading C1. The 'unload' action was triggered by *DecideAction* which used *AgentState* artifacts indicating that the agent 'has a civilian' and is 'at a refuge'. In turn, the *AgentState* indicating that the agent was at a refuge was generated by a 'move' action that used a path artifact generated by *PlanPath* that was triggered by *DecideAction*. Note that the kernel is required only in RCRS and not in the real world, as humans do not need a 'kernel' to tell them the results of their actions.

Querying RCRS Provenance Understanding why events occurred in RCRS and how decisions affected them requires mapping provenance questions into provenance queries. Querying consists of traversing the OPM DAG to produce a provenance graph pertaining to the data items of interest. A query is formed of a *query data handle*, which identifies the entity for which the provenance is sought, and the *scope* of traversal [13], which identifies what forms a relevant answer to the query (i.e. what parts of the OPM graph are of interest to the querier).

Traversing a graph produced by RCRS should exploit *RobocupProfile*'s characteristics. For instance, the data handle can specify the type of artifact pertaining to the data item for which provenance is sought, e.g. for the query of question 1 in §4, the data handle is identified by the type of the artifact, namely *TaskResult*. Also, the scope can identify which paths to prune by discarding certain sub-types of nodes or edges, as well as stopping the traversal when certain types of nodes and edges are reached. Additionally, accounts can be used to prune nodes and edges that are not in the scope, e.g. nodes belonging to *KernelAcct* can be pruned when traversing the graph to address question 1.

We now briefly show how to address the questions in §4. Though all are queried based on the above, each has varied aspects and is handled distinctly.
Answering question 1 is done by finding and traversing a series of *AmbulanceAction*s where the last one generates a *TaskResult* concerning C1.
Question 2 requires checking why each agent did not save civilian C2, i.e. why C2 never became their priority on their sorted list of civilians. We use Chapman and Jagadish's algorithms which explore why certain data item were not returned by a query [5] and we find the *SortedListCivs* where C2 does not show.
Question 3 requires finding *AgentState*s concerning both civilians and their *UpdatedState* dependencies, and if needed the artifacts they were derived from.
Question 4 considers the activity of question 1 (sequence of actions taken to save a civilian) and analyzes its beginning, ending, and the number of steps between them. By showing the number of processes that took place within the activity, we point out the factors that contributed to its elongation.

6 Related Work

The Belief-Desire-Intention framework [20] is the best known and best studied model in the Agent Theories, Architectures, and Languages community [8]. However, it lacks mechanisms that allow agents to learn based on past experiences, thus it has been extended to allow the use of learning in [19]. Other recent work has aimed to make use of history and experiences by using an agent's past experiences and its history of interactions with other agents. While the aim of [12] is to improve organizational performance by presenting a structural adaptation method that is based on the history of interactions with agents to be used by an agent to self-organize and decide to drop relations with other agents; most of the other work is centered on using past interactions with other agents so that an agent can ultimately choose whether or not to trust, cooperate with, and rely on those other agents [23,7,4,10]. The aforementioned work does not treat past experiences as provenance data and so does not exploit any provenance framework or model nor does it utilize the history for the benefits of understanding what went on and why. This is done in [11,15] where distributed processes in an organ transplant management application are treated as agents and the provenance of their actions and interactions is recorded. Specifically, a provenance model that extends PrIMe to capture the goals and intentions of agents in distributed systems is presented in [15]. Although we apply our approach to multi-agent systems which fall under the umbrella of distributed systems, our focus is on how and why decisions were made; and *RobocupProfile* considers agents' goals and intentions through their influences on the decision making process.

Approaches to explanations in rule-based systems like the expert-system MYCIN included paraphrasing the system code; however, such expert-systems do not provide justifications for their rules [18]. On the other hand, the generation of explanations benefits from decision theory as a powerful tool for justifying decisions [18], commonly used in the contexts of decision trees and reasoning about preferences. Nevertheless, such approaches would be limited when reasoning about causality and chains of decisions; at least not to the extent that the application of provenance provides.

Additionally, addressing accountability of the autonomous entities forming distributed systems is a challenge [16]. For users to have confidence in them, these systems must be made accountable, i.e. be enabled to prove their compliance with policies [25]. Several approaches use provenance to make systems accountable, including [6] and [21]. Finally, the need to secure provenance is vital in many critical areas such as law, scientific data, and authorship [9] and would also be important in the decision making domain. Addressing this need includes securing provenance [3] and maintaining provenance integrity [9].

7 Conclusion and Future Work

In summary, our work provides a proof-of-concept for how provenance can be used to track decisions in automated emergency response systems. We presented the use case "Provenance of Decision Making in Emergency Response" as the motivation for our work. RCRS was used as a testbed application and PrIMe was applied to it to make it provenance-aware. Furthermore, OPM was used to produce provenance DAGs, making use of a novel OPM profile that specializes OPM and conceptualizes decisions in the context of emergency response. Thus, the presented work provides a means for justifying automated decisions in emergency response systems by capturing why and how they were made.

Our work shows how provenance of decisions can be exploited in an offline manner, after an application has terminated, to understand automated decisions in complex scenarios. We believe that provenance of decisions can also be used in an online manner for the purpose of making better decisions. This is especially true when previous decisions need to be revised because some new observation has invalidated or complemented previous knowledge. We have proposed these scenarios in our use case and our future work will aim to address these points.

Acknowledgements

This research was undertaken as part of the ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Systems) Project and is jointly funded by a BAE Systems and EPSRC strategic partnership (EP/C 548051/1).

References

1. : (RoboCup Rescue) `http://www.robocuprescue.org/index.html`.
2. Bose, R., Frew, J.: Lineage retrieval for scientific data processing: a survey. ACM Computing Surveys (**37**) 1–28
3. Braun, U., Holland, D.A., Muniswamy-Reddy, K.K., Seltzer, M.I.: Coping with cycles in provenance. Technical report, Harvard University (2006)

4. Chalkiadakis, G., Boutilier, C.: Sequential decision making in repeated coalition formation under uncertainty. In: Proc. of AAMAS 2008, (Estoril, Portugal)
5. Chapman, A., Jagadish, H.V.: Why not? (In: Proc. of the 35th SIGMOD int'l conf. on Management of data)
6. Chorley, A., Edwards, P., Preece, A., Farrington, J.: Tools for tracing evidence in social science. (In: Third Int'l Conf. on e-Social Science, October 2007)
7. Fullam, K.K., Barber, K.S.: Dynamically learning sources of trust information: experience vs. reputation. (In: Proc. of AAMAS 2007)
8. Georgeff, M., Pell, B., Pollack, M., Tambe, M., Wooldridge, M.: The Belief-Desire-Intention Model of Agency. In: Proc. of the 5th Int'l Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages. (1998)
9. Hasan, R., Sion, R., Winslett, M.: The case of the fake picasso: Preventing history forgery with secure provenance. (In: Proc. of 7th USENIX Conference on File and Storage Technologies, FAST 2009) 1–14
10. Khosravifar, B., Gomrokchi, M., Bentahar, J., Thiran, P.: Maintenance-based trust for multi-agent systems. In: Proc. of AAMAS 2009, (Budapest, Hungary)
11. Kifor, T., Varga, L.Z., Vazquez-Salceda, J., Alvarez, S., Willmott, S., Miles, S., Moreau, L.: Provenance in agent-mediated healthcare systems. IEEE Intelligent Systems **21** (2006) 38–46
12. Kota, R., Gibbins, N., Jennings, N.R.: Self-organising agent organisations. In: Proc. of AAMAS 2009, (Budapest, Hungary)
13. Miles, S.: Electronically querying for the provenance of entities. In: Third Int'l Provenance and Annotation Workshop. Vol. 4145. Springer (2006) 184–192
14. Miles, S., Groth, P., Munroe, S., Moreau, L.: Prime: A methodology for developing provenance-aware applications. ACM TOSEM (2010)
15. Miles, S., Munroe, S., Luck, M., Moreau, L.: Modelling the provenance of data in autonomous systems. (In: Proc. of AAMAS 2007)
16. Moreau, L.: The foundations for provenance on the web. Foundations and Trends in Web Science (In Press) (2010)
17. Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y.L., Stephan, E., Van Den Bussche, J.: The Open Provenance Model Core Specification (v1.1). Future Generation Computer Systems (In Press) (2010)
18. Papamichai, K.N., French, S.: Explaining and justifying the advice of a decision support system: a natural language generation approach. Expert Systems with Applications **24** (2003)
19. Phung, T., Winikoff, M., Padgham, L. In: Learning Within the BDI Framework : An Empirical Analysis. (2005) 282–288
20. Rao, A.S., Georgeff, M.P.: BDI Agents : From Theory to Practice. In: Proceedings of the First International Conference on MultiAgent Systems. (1995)
21. Ringelstein, C., Staab, S.: PAPEL : A Language and Model for Provenance-Aware Policy Definition and Execution. In: BPM 2010, (Hoboken, NJ, USA)
22. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance in e-science. SIGMOD Rec. **34** (2005) 31–36
23. Teacy, W.T., Patel, J., Jennings, N.R., Luck, M.: Travos: Trust and reputation in the context of inaccurate information sources. Autonomous Agents and Multi-Agent Systems **12** (2006) 183–198
24. W3C Provenance Incubator Group: (Provenance dimensions) `http://www.w3.org/2005/Incubator/prov/wiki/Provenance_Dimensions` last accessed 01-Mar-2010.
25. Weitzner, D.J., Abelson, H., Berners-Lee, T., Feigenbaum, J., Hendler, J., Sussman, G.J.: Information accountability. Commun. ACM **51** (2008) 82–87