Separating representation, reasoning, and implementation for interaction management

Separating representation, reasoning, and implementation for interaction management

Mary Ellen Foster and Ronald P. A. Petrick

A fundamental component of any dialogue system is the *interaction manager* [4], whose primary task is to carry out *action selection*: that is, based on the current state of the interaction and of the world, the interaction manager makes a high-level decision as to which spoken, non-verbal, and task-based actions should be taken next by the system as a whole. In contrast to more formal, descriptive accounts of dialogue (e.g., [1]), which aim to model the full generality of language use, work on interaction management has concentrated primarily on developing end-to-end systems and on evaluating them through interaction with human users [7].

A number of toolkits are available to support the construction of such end-to-end dialogue systems [2, 3, 8, 9, 13, 14, 17]. Such toolkits generally incorporate three main features. First, they provide a representational formalism for specifying states and actions. Second, the state/action representation is usually tightly linked to the reasoning strategy that is used to carry out action selection. Finally, most include a technical framework for modular system development. While these features clearly simplify the task of implementing individual end-to-end systems, the fact that the features are so tightly connected complicates the task of comparing representational formalisms or reasoning strategies: in general, to carry out such a comparison, there is no alternative but to re-implement the entire system across multiple frameworks [e.g., 11].

The overall problem of selecting high-level actions for an intelligent agent is not unique to dialogue systems, but is a problem addressed in a variety of research communities including *automated planning*. In automated planning, the emphasis is on applying problem-solving techniques to find an ordered sequence of actions (a

M. E. Foster
School of Computing Science, University of Glasgow, Glasgow G12 8RZ, Scotland, UK, e-mail: MaryEllen.Foster@glasgow.ac.uk

R. P. A. Petrick
School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, Scotland, UK, e-mail: rpetrick@inf.ed.ac.uk

plan) that, when chained together, will transform an initial state into a state where a set of specified goal objectives are achieved.[1]

One important feature of much planning research is that the tools developed by this community usually support one of a number of common representation languages such as PDDL [10], PPDDL [18], or RDDL [16], among others. Many of these languages have been developed or extended as part of the International Planning Competitions (IPC) [5, 6], which have run approximately every other year since 1998 within the context of the International Conference on Automated Planning and Scheduling (ICAPS). Even when a planner implements its own representation language, which may differ (usually syntactically) from the standard planning languages, additional work is often performed to establish the relationship between such languages and the more common representations.

These activities have led to important benefits for the planning community. First, by adopting common representations, the task of modelling a planning domain can be separated from the task of implementing an efficient engine for solving problems in those domains. This allows different planning engines to be developed and directly compared, either quantitatively or qualitatively, on a common set of inputs (i.e., planning problems). Second, planning domains and planning engines can be shared, leading to the development of common benchmarks for future planning systems, as well as an improvement in the baseline systems that can solve problems in these domains. Indeed, the success of the IPC has produced an extensive collection of planning domains which have been used for such tasks.[2] Finally, the representation languages themselves—and the planning problems they support—can be studied and compared, leading to a better understanding of the complexity of particular classes of domains and problems [15], and the tradeoffs of using one language over another.

Within the dialogue systems community, the diversity of approaches has had the result that, while it is common to compare interaction management strategies within a single framework, it is relatively uncommon to compare the representational ability and reasoning performance across different frameworks. (One exception is [11] which carried out such a comparison, but which required the entire dialogue system to be implemented separately in each formalism.) As a result, we believe that the experiences of the automated planning community could be applied within the dialogue systems community, leading to similar positive results—and closer links between these two research fields.

Acknowledgements

This research was supported in part by the European Commission's Seventh Framework Programme through grant no. 270435 (JAMES, `james-project.eu`) and grant no. 610917 (STAMINA, `stamina-robot.eu`).

[1] This differs somewhat from the task of interaction management, where the goal is (usually) to find the next system action, rather than a complete action sequence. However, note that a system that is able to achieve the latter can also be used in the former context (see, e.g., [12].)

[2] See `http://www.icaps-conference.org/index.php/Main/Competitions`.

References

1. Asher N, Lascarides A (2003) Logics of Conversation. Cambridge University Press
2. Bohus D, Rudnicky AI (2009) The RavenClaw dialog management framework: Architecture and systems. Computer Speech & Language 23(3):332–361, doi:10.1016/j.csl.2008.10.001
3. Bos J, Klein E, Lemon O, Oka T (2003) DIPPER: Description and formalisation of an information-state update dialogue system architecture. In: Proceedings of SIGdial 2003, pp 115–124
4. Bui TH (2006) Multimodal dialogue management - state of the art. Tech. Rep. 06–01, University of Twente (UT), Enschede, The Netherlands
5. Coles A, Coles A, García Olaya A, Jiménez S, Linares López C, Sanner S, Yoon S (2012) A survey of the seventh international planning competition. AI Magazine 33(1):83–88, doi:10.1609/aimag.v33i1.2392
6. ICAPS (2015) ICAPS competitions. `http://www.icaps-conference.org/index.php/Main/Competitions`
7. Jokinen K, McTear M (2009) Spoken dialogue systems. Synthesis Lectures on Human Language Technologies 2(1):1–151, doi:10.2200/S00204ED1V01Y200910HLT005
8. Larsson S, Traum DR (2000) Information state and dialogue management in the TRINDI dialogue move engine toolkit. Natural Language Engineering 6(3&4):323–340, doi:10.1017/S1351324900002539
9. Lison P (2015) A hybrid approach to dialogue management based on probabilistic rules. Computer Speech & Language doi:10.1016/j.csl.2015.01.001
10. McDermott D, Ghallab M, Howe A, Knoblock C, Ram A, Veloso M, Weld D, Wilkins D (1998) PDDL – The Planning Domain Definition Language (Version 1.2). Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control
11. Peltason J, Wrede B (2011) The curious robot as a case-study for comparing dialog systems. AI Magazine 32(4):85–99, doi:10.1609/aimag.v32i4.2382
12. Petrick RPA, Foster ME (2013) Planning for social interaction in a robot bartender domain. In: Proceedings of ICAPS 2013, Special Track on Novel Applications
13. Rich C, Sidner CL (1998) COLLAGEN: A collaboration manager for software interface agents. User Modeling and User-Adapted Interaction 8(3–4):315–350, doi:10.1023/A:1008204020038
14. Rich C, Sidner CL (2012) Using collaborative discourse theory to partially automate dialogue tree authoring. In: Intelligent Virtual Agents, Lecture Notes in Computer Science, vol 7502, pp 327–340, doi:10.1007/978-3-642-33197-8_34
15. Rintanen J (2004) Complexity of planning with partial observability. In: Proceedings of ICAPS 2004, pp 345–354
16. Sanner S (2010) Relational dynamic influence diagram language (RDDL): Language description. `http://users.cecs.anu.edu.au/~ssanner/IPPC_2011/RDDL.pdf`
17. Skantze G, Al Moubayed S (2012) IrisTK: A statechart-based toolkit for multi-party face-to-face interaction. In: Proceedings of ICMI 2012, pp 69–76, doi:10.1145/2388676.2388698
18. Younes HLS, Littman ML (2004) PPDDL1.0: An extension to PDDL for expressing planning domains with probabilistic effects. Technical Report CMU-CS-04-162, Carnegie Mellon University