Interaction Design Patterns for Adaptive Human-Agent-Robot Teamwork in High-Risk Domains

Article · July 2016

DOI: 10.1007/978-3-319-40030-3_22

3 authors, including:

Mark Neerincx
TNO
294 PUBLICATIONS **2,068** CITATIONS
SEE PROFILE

Jurriaan van Diggelen
TNO
53 PUBLICATIONS **382** CITATIONS
SEE PROFILE

Interaction Design Patterns for Adaptive Human-Agent-Robot Teamwork in High-Risk Domains

Mark A. Neerincx(✉), Jurriaan van Diggelen, and Leo van Breda

TNO, Kampweg 5, 3769 DE Soesterberg, Netherlands
{mark.neerincx, jurriaan.vandiggelen, leo.vanbreda}@tno.nl

Abstract. Integrating cognitive agents and robots into teams that operate in high-demand situations involves mutual and context-dependent behaviors of the human and agent/robot team-members. We propose a cognitive engineering method that includes the development of Interaction Design patterns for such systems as re-usable, theoretically and empirically founded, design solutions. This paper presents an overview of the background, the method and three example patterns.

Keywords: Design patterns · Human-Agent teamwork · Cognitive robots · Cognitive engineering · High-Risk domain

1 Introduction

A clear need exists for the deployment of robots to establish effective and safe operations in high risk domains, e.g. for firefighting, search and rescue, and defense. To meet this need, research in the field of supervisory control provided advanced multi-modal user interfaces for the robot operators, focusing on dedicated, small human-robot settings (e.g., two operators that control an Unmanned Aerial Vehicle, UAV, or a single operator that (tele-)operates an Unmanned Ground Vehicle, UGV). To enhance the robot's functions and usability, its level of automation has been increased, so that the operator's role became more supervisory in nature, overseeing the automated activation of programmed events (e.g., making sure the appropriate event is activated at the appropriate time) and managing unexpected changes to the automated mission plan. Associated operator interfaces for the robots have been developed that take into account issues associated with automation management, including vigilance, attention management, clumsy automation, etc. Subsequently, next-generation multiple-robot systems have been developed that can be supervised and controlled by a single supervisor at a higher abstraction level, due to system's increased capability to make 'lower level decisions'. For the supervisory control of single and multiple robots, inventories of critical human factors issues were made, e.g. on situation awareness, workload, performance and safety. Standard operator interface design guidelines associated with supervisory control were developed to facilitate interoperability across (semi) autonomous platforms, and

D. Harris (Ed.): EPCE 2016, LNAI 9736, pp. 211–220, 2016.
DOI: 10.1007/978-3-319-40030-3_22

for identifying, prioritizing, and addressing human factors challenges associated with robot supervisory control [4, 11, 19, 20, 22, 26].

However, the content and scope of these guidelines fail to address current operational demands and to steer the required developments and applications of robotics and artificial intelligence. The supervisory control paradigm still regards robots as "obedient servants" that only do work after they have been explicitly told to do so by a human operator, which could cause an unacceptably high workload. Furthermore, the human is not necessarily the best decision maker, as the robot may possess information which is unknown to the operator. As the amount of robots (or UxV's) is increasing and these robots are being employed in a wider variety of tasks, they should become more proactive in their behavior than in the supervisory control paradigm. To realize this, we should aim at robots as team-members [6]. A major challenge for such an approach is to integrate the (intelligent) robots into the dynamic teamwork in such a way that the robots complement human capabilities, relieve them from demanding tasks (e.g., observation, reconnaissance, search, securing and sampling) and do not pose additional demands on them (cf., [18]). That is, we aim at the development of robots that become more and more able to act as adaptive team-members (e.g., by sharing knowledge, pursuing team goals, and coordinating "own" actions with actions of others). In our approach, such a system encompasses networked Humans, Agents and Robots, which show Teamwork (HART) in a "smart environment" (i.e., networked interactive things and knowledge bases; see Fig. 1). The agents support goal-oriented behaviors, driven by task, context, team and user models [3, 12], and the ontology provides the knowledge representations to establish joint knowledge-based behaviors (based on shared mental models, transactive memory systems, and shared situation awareness; [10]).

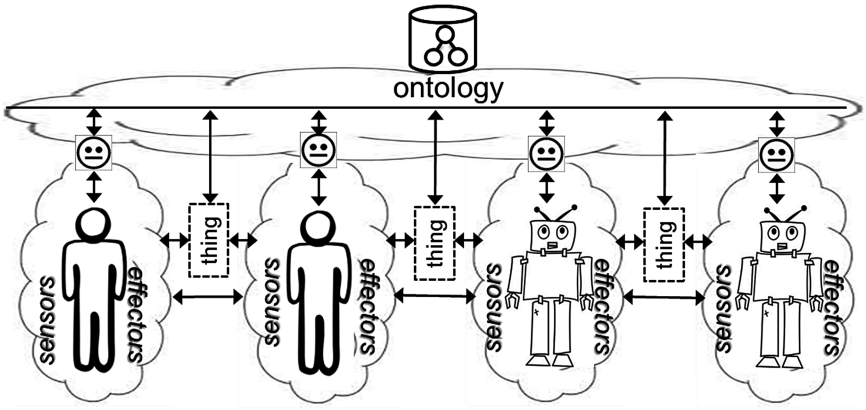


Fig. 1. Human-Agent-Robot Teamwork (HART) in a smart environment

As the behaviors of the humans, agents and robots are adaptive (i.e., towards one another and towards the dynamic outside), design and implementation of the optimal set of behaviors is intrinsically complex. Whereas several methodologies for agent-based software engineering exist (e.g. [13]), these methodologies focus on systems that consist

completely of software agents and robots, and do not consider the human interactions that is required in HART. To fill this gap, we propose to use Interaction Design (ID) patterns as an integral part of cognitive engineering, addressing the mutual dependent human, agent and robot behaviors in an explicit Interaction Design Rationale. These ID-patterns (1) justify the design choices with theoretical and empirical foundations, (2) show the similarities between different instantiated interaction designs and (3) may be put into a library of reusable (justified) HART ID-patterns.

2 Design Patterns

Alexander [1] was the originator of the pattern concept, defining it as a description of "[…] a problem which occurs over and over again in our environment, and […] the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice". His philosophy of constructive, coherent and meaningful design in architecture, inspired the development of pattern languages in many other domains and application fields. Examples are Workflow Patterns [27]), User Interface Design Patterns [25, 28], Interaction Design [5], Design Patterns for sociality in human-robot interaction [14], human-computer interaction [7], patterns to manage software complexity [9], and patterns for collaborative technology [24].

For our purpose of explicating the HART Design Rationale during research and development, the following characteristics are important. A pattern is a structured description of an invariant solution to a recurrent problem within a context. It abstracts true interactions, is generative and includes notion of temporality. The HART ID-patterns should capture good practice and provide theoretical account, i.e., they (1) represent "big ideas" with their design rationale, (2) reflect design values, (3) contain common concepts to communicate the design rationale (as a "lingua franca"), (4) are grounded in the domain and include examples, and (5) have different levels of abstraction and scales.

3 Pattern Engineering

We propose to integrate the ID-pattern development process into a general situated Cognitive Engineering methodology that derives a coherent base-line of use cases, requirements and claims from work, domain, human factors and technology analyses [21]. This baseline describes the *what* (requirements), *when* (use cases) and *why* (claims) of the design, whereas the patterns describe *how* the human-agent interaction will take place [17]. These interaction patterns are generalization of specific user interface and dialogue instantiations (the interdependent multi-actor "look-feel-and-hear").

For HART patterns, we distinguish the following key concepts [23]:

- ***Actor***: In a HART system the actor can be *Human*, *Agent* or *Robot*. Note that we use the term "Actor", where Schulte et al. [23] use the term "Worker". Actor refers to "activity" instead of work and is as such a more general term. In this way, we can describe generic patterns on joint human-agent/robot activities that take place within and outside work organizations (e.g. informal caregiving).

- ***Relationship***: The relationship between actors can be *Supervisory* and/or *Collaborative*. These two parameters are similar to the distinction of Schulte et al. [23] between hierarchical and heterarchical relationships. However, we distinguish the main concept "Relationship" to enable the creation of patterns that adjust such relationships during the work processes.
- ***Location***: Actors can perform their work at the *Same* (co-location) or a *Distant* (distributed) location. Also here, we are focusing on the dynamics, e.g., patterns that describe agent support for " roaming operators" (see Sect. 4.2).
- ***Pattern status***: the status of the pattern can be *Proto* (i.e., in construction) or *Grounded* (e.g., empirically validated in an experiment).

Pattern engineering aims at the generation, sharing, use and evolution of design knowledge [16]. To make progress in the field of HART, the research and design can make use of available patterns and anticipate for the refinement or construction of relevant patterns, taking the following steps:

1. Identify key design problems
2. Search for available design patterns
3. If no pattern can be found, and if it is a general, recurrent design problem:
   - Start with a *Proto Pattern*[1], a pattern "in construction", i.e., a design problem and solution documented in a pattern form (yet lacking empirical grounding)
4. Provide different instantiations (examples)
5. Test, refine and validate these examples
6. If successful:
   - Make the Design Pattern accessible in library (of best practices)

4 Example ID-Patterns

This section presents briefly three example ID-patterns to share HART research progress: for making working agreements, for establishing adequate human supervision of the delegated tasks and for anticipating required colocation actions.

4.1 Adjustable Human-Agent Working Agreements

Our first example focuses on the enabling of making adjustable working agreements between humans and agents to establish the required adaptability of the teamwork. This pattern aims to overcome a common problem present in most modern SCADA systems (Supervisory Control And Data Acquisition), where either the system behaves fully autonomously, or where the full control is allocated to the human. Using an agent and this design pattern, a third option is introduced which supports dynamic and adaptive human-agent (sub)task allocation (i.e., the SCADA system is evolving into a HART system). For specific work contexts, the human can set agreements with the agent on how the tasks will be allocated.

[1] cf. http://c2.com/cgi/wiki?ProtoPattern.

Title	Obtain adjustable human-agent working agreements on object handling

Design Problem

- The information that the team has to process for object handling is highly dynamic, causing high peaks and deep hollows in workload
- The agent can perform "only" well-delineated information processing tasks adequately
- The human operator should remain in the loop (1) to maintain the Situation Awareness for adequate performance of the complex or unsteady tasks, (2) to assess the work strategy, and (3) to maintain overall responsibility

Design solution

The *human* operator can choose the conditions and rules for delegating tasks to the agent with the objectives to harmonize operator workload and maintain situation awareness (SA) under the dynamic conditions for optimal team performance. The human-agent relationship is *supervisory* and *collaborative*. The human operator, who is in charge of the work process (as "Creditor"), sets (1) the delegation criteria (e.g., the area of operation and characteristics of possible objects in this area that the agent is obliged to handle) and (2) the corresponding handling tasks (e.g., identify and monitor). As an electronic partner, the agent will meet this social commitment and act according to the defined settings with the corresponding obligations, i.e., as "Debtor", show the task outcomes with the conditions that led to these outcomes (e.g., sensed attributes of an object). In this way, agents can be committed to provide an advice on the handling of specific (critical) objects, while handling (standard) objects themselves. A formal language has been developed to implement such commitments in a human-agent system [15].

Use when

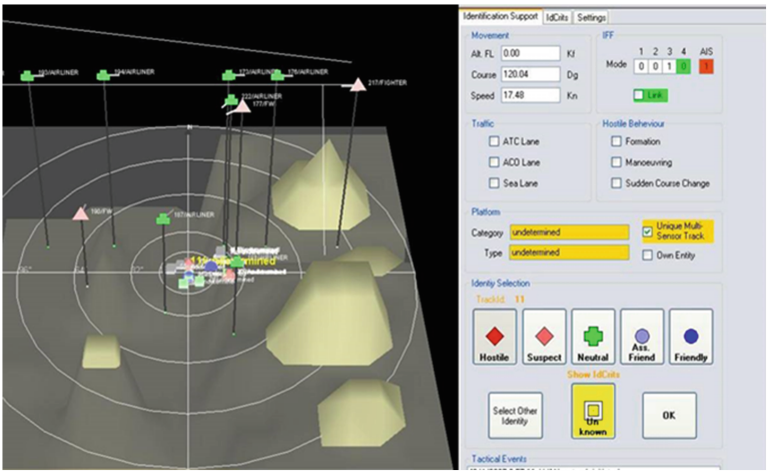
When a clear-cut part of the work can be automated and humans are needed (a) to deal with the uncertain or ill-defined information, and (b) to take the overall responsibility for the information-processing strategy and outcome.

Design rationale

HART-system behaviors are prompted and constrained by norms (e.g., flight regulations). To address situational dynamics, the responsibilities for specific task objects (e.g., tracks) are actively divided by the human operator by specifying the range of distinctive object attributes (e.g., flight height, region, …) that the agent has to process (i.e., is obliged to identify). The human operator can maintain the overall responsibility of the work well, because (1) she initiates and may always adjust the agreement, and (2) the agent's behavior is transparent (i.e., showing the current object attributes and rules that led to the identification outcome). In other words, the agent will act according to the agreement settings with the corresponding obligations and, as "Debtor", show the conditions that led to the task outcomes (e.g., speed, direction, height and distance of an object, which led to agent's identification "neutral"). In general, the human takes the more difficult objects and the agent is left with the easy ones (e.g., neutral objects that take an air or sea way and identify themselves).

Example

A prototype of a "track handling agent" for a naval operator was developed [2] and tested [8]. The general task is to assess tracks in the environment of the ship and decide whether it is "unknown", "friendly", "assumed friendly", "neutral", "suspect" or "hostile". These tracks are the domain-related instantiations of objects for which a number of tasks can be executed (e.g., classification, identification, guidance and/or engagement).



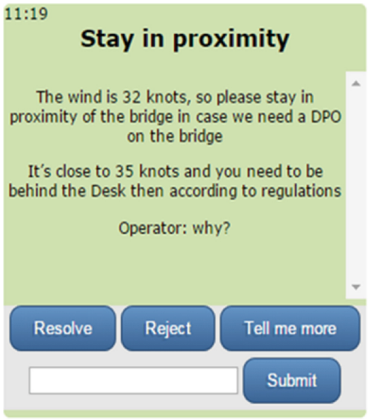
Via color- and form-coded icons, the source (human or agent) and outcome of the identification is shown in the display. Object attributes are shown when clicking on the corresponding icon. A timeline shows the tasks that have been and are being performed for the operator and the agent. In a separate tab, the work agreement settings can be specified and edited. For example, for the identification task, the operator can set a speed threshold of 400 mph. If higher than 400, the agent has to give an advice on the identity of the track, whereas below 400 the agents sets the identity itself. This granular work distribution with object attributes matches closely humans way of dynamic work distribution (e.g., when the number of signals, the response times to these signals, and/or the workload are high).

Status

Grounded [8]: The effects of the "track handling agent" was evaluated in a high-fidelity command & control setting with eight naval officers. The overall efficiency increased with 60%, particularly for the complex scenarios (65%). Downsides of the agent did not appear.

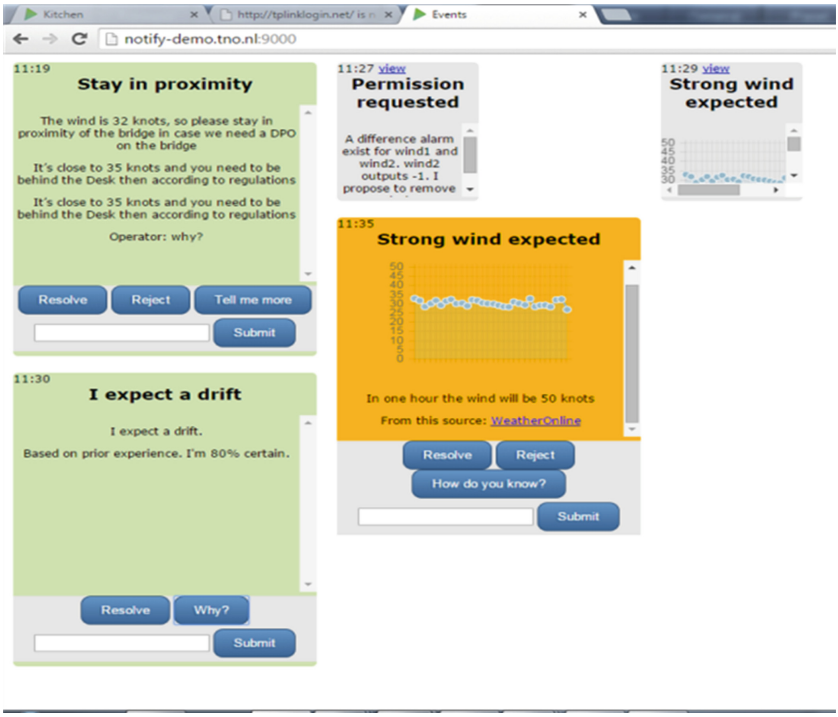
4.2 Transfer from Distant to Co-location

The second design pattern aims at providing a solution for the problem that human control has some context requirements which must be fulfilled before control can be passed to the human. One of these context requirements is spatial location. For example, when the system operates in fully autonomous mode, and a problem occurs, the human operator should be able to make it back to the workstation within a certain time limit.

Title	Demand operator to stay in vicinity of workstation.
Design Problem The agent predicts that human intervention might be necessary soon, which cannot be done from a mobile device. It asks the operator to stay in the vicinity of the workstation.	
Design solution Popup window with short explanation of the type of expected problems and time frame. The operator can ask the agent for more explanation, and decide to agree or disagree to stay in the vicinity.	
Use when Agent expects to switch from autonomous mode to a semi-autonomous mode which requires a stationary operator.	
Design rationale Operator is more likely to follow the system's advice to stay in the vicinity if (s)he understand why this is necessary.	
Example In the control room of a ship with Dynamic Positioning, an agent that provides proximity notification allows operators to roam about the vessel. This notification provides a summary of the proximity need. The operator can resolve, reject, annotate or ask for more information.  	
Status	*Proto*

4.3 Management of Interaction Processes

The interaction design patterns such as the ones described above have been designed to realize sensible human-agent interaction by themselves. This does not guarantee that the human can cope with multiple interactions running simultaneously. The third design pattern that we will discuss aims to solve that problem.

Title	Manage multiple interactions between user and system
Design Problem When many interactions with the agent are required simultaneously, the user gets overloaded with information.	
Design solution A container window which contains all separate interactions as separate tiles. The important interactions are shown intrusively (i.e. in color and large), and the less important interactions are shown non-intrusive (smaller and greyed out). The container shows the most important 7 windows in an intrusive way. The user can choose to dismiss any interaction as non-important using the “resolve” button.	
Use when Multiple different types of interactions are required simultaneously.	
Design rationale By limiting the amount of intrusive interactions to seven, human operators are capable of processing them simultaneously.	
Example 	
Status	*Proto*

5 Conclusions

Integrating cognitive agents and robots into teams that operate in high-demand situations involves mutual and context-dependent behaviors of the human and agent/robot team-members. Figure 1 shows the concept of Human-Agent-Robot Teamwork (HART), encompassing agent- and ontology-mediated human-robot collaboration in order to establish adaptive teamwork. We propose a cognitive engineering method that includes the development of Interaction Design patterns for such systems as re-usable, theoretically and empirically founded, design solutions. These patterns are used to explicate and share HART research and development results. In this way, a pattern for making working agreements has been constructed and tested. For establishing adequate human supervision of the delegated tasks, an approval-request pattern was constructed, and for anticipating required colocation actions, a vicinity-advice patterns was constructed. These patterns can be instantiated for different use cases and into specific interaction designs, in order to realize the required adaptive human-robot teamwork.

Acknowledgements. This research is supported by the EU FP7 project 609763 (TRADR), and the TNO Defense research program V1340 on Unmanned Systems.

References

1. Alexander, C., Ishikawa, S., Silverstein, M.A.: Pattern Language: Towns, Buildings, Construction. Center for Environmental Structure series. Oxford University Press, Berkeley (1977)
2. Arciszewski, H.F., De Greef, T.E., Van Delft, J.H.: Adaptive automation in a naval combat management system. IEEE Trans. Syst. Cybern. A Syst. Hum. **39**(6), 1188–1199 (2009)
3. Bagosi, T., de Greeff, J., Hindriks, K.V., Neerincx, M.A.: Designing a Knowledge Representation Interface for Cognitive Agents. In: Baldoni, M., Baresi, L., Dastani, M. (eds.) EMAS 2015. LNCS, vol. 9318, pp. 33–50. Springer, Heidelberg (2015). doi:10.1007/978-3-319-26184-3_3
4. Banbury, S., Gauthier, M., Scipione, A., Hou, M.: Intelligent adaptive systems: literature research of design guidance for intelligent adaptive automation and interfaces. Defence Research and Development Canada - Toronto, CR 2007-075 (2007)
5. Borchers, J.O.: A pattern approach to interaction design. AI SOC. **15**(4), 359–376 (2001)
6. Bradshaw, J.M., Feltovich, P., Johnson, M., Breedy, M., Bunch, L., Eskridge, T., Jung, H., Lott, J., Uszok, A., van Diggelen, J.: From Tools to Teammates: Joint Activity in Human-Agent-Robot Teams. In: Kurosu, M. (ed.) HCD 2009. LNCS, vol. 5619, pp. 935–944. Springer, Heidelberg (2009)
7. Dearden, A., Finlay, J.: Pattern languages in HCI: a critical review. Hum. Comput. Inter. **21** (1), 49–102 (2006)
8. De Greef, T.E., Arciszewski, H.F.R., Neerincx, M.A.: Adaptive automation based on an object-oriented task model: implementation and evaluation in a realistic C2 environment. J. Cognitive Eng. Decis. Mak. **4**, 152–173 (2010)
9. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addinson-Wesley, Reading (1995)

10. Gevers, J.M.P., Uitdewilligen, S., Margarida Passos, A.: Dynamics of team cognition and team adaptation: introduction to the special issue. Eu. J. Work Organ. Psychol. **24**(5), 645–651 (2015)
11. Harrison, J.A., Forster, M.J.: Human systems integration requirements in systems acquisition. In: Booher, H.R. (ed.) Handbook of Human Systems Integration (2003). Wiley, Hoboken (2003)
12. Hindriks, K.V.: Programming rational agents in goal. In: El Fallah Seghrouchni, A., Dix, J., Dastani, M., Bordini, H.R. (eds.) Multi-agent Programming, pp. 119–157. Springer, New York (2009)
13. Jennings, N.R.: An agent-based approach for building complex software systems. Commun. ACM **44**(4), 35–41 (2001)
14. Kahn, P.H., Freier, N.G., Kanda, T., Ishiguro, H., Ruckert, J.H., Severson, R.L., Kane, S.K.: Design patterns for sociality in human-robot interaction. In: Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction, pp. 97–104. ACM (2008)
15. Kayal, A., Brinkman, W.P., Neerincx, M.A. van Riemsdijk, M.B.: A social commitment model for location sharing applications in the family domain. Int. J. Hum. Comput. Stud. (to appear)
16. Kohls, C.: The theories of design patterns and their practical implications exemplified for e-learning patterns. Dissertation, Katolischen Universität Eichstätt-Ingolstadt. Eichstätt (2013)
17. Mioch, T., Ledegang, W., Paulissen, R., Neerincx, M.A., Van Diggelen, J.: Interaction design patterns for coherent and re-usable shape specifications of human-robot collaboration. In: EICS 2014 (Rome, Italy, June 17–20), pp. 75–83. ACM (2014)
18. Murphy, R.R.: Disaster Robotics. MIT Press, Cambridge (2014)
19. NATO. The NATO Unmanned Aircraft Systems Human Systems Integration Guidebook (2012)
20. NATO. The NATO Human View Handbook (2007)
21. Neerincx, M.A.: Situated cognitive engineering for crew support in space. Pers. Ubiquitous Comput. **15**(5), 445–456 (2011)
22. Reising, J. (ed.): Uninhabited Military Vehicles (UMVs): Human factors Issues in Augmenting the Force. NATO RTO technical report RTO-TR-HFM-078. NATO, Brussel (2009)
23. Schulte, A., Donath, D., Lange, D.S.: Design patterns for human-cognitive agent teaming. In: 13th Conference on Engineering Psychology & Cognitive Ergonomics. LNCS. Springer
24. Schümmer, T., Lukosch, S.: Patterns for Computer-Mediated Interaction. Wiley, Chichester (2007)
25. Tidwell, J.: Designing Interfaces. O'Reilly Media, Inc., Sebastopol (2010)
26. Van Breda, L. (ed.): Supervisory Control of Multiple Uninhabited Systems – Methodologies and Enabling Human-Robot Interface Technologies. NATO RTO technical report AC/323 (HFM-170)TP/451. NATO, Neuilly-sur-Seine Cedex, France (2012). ISBN:978-92-837-0167-5
27. Van der Aalst, W., Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow patterns. Distributed Parallel Databases **14**(1), 5–51 (2003)
28. Van Welie, M., Van der Veer, G.C.: Pattern languages in interaction design: structure and organization. In: Proceedings of Interact 2003, Zürich, Switzerland, pp. 527–534. IOS Press, Amsterdam, 1–5 September 2003