Using Stories to Deepen Shared Human-Computer Understanding of PMESII Systems

Robert E. Wray, Jeremiah T. Folsom-Kovarik, Dylan Schmorrow, Randolph M. Jones and Robert Marinier

Abstract Humans tend to represent and to understand the world in terms of stories, while computer reasoning tends to require formal, mathematical representations. This paper describes a research prototype that enables computers to parse human stories and use collections of those stories to inform causal modeling of political, military, economic, social, infrastructure, and information (PMESII) systems. We introduce the need for causal modeling, the approach we have taken in implementing an initial proof-of-concept and the results from pilot testing of the software that illustrates functional capabilities and opportunities for deepening story-based computer interpretation of stories.

Keywords PMESII modeling · Computational narrative · Human-systems integration

1 Introduction

A gap, sometimes a gulf, typically exists between human understanding and computer understanding of a current situation, a past event, or a simulation. The gap leads to misunderstandings, mistakes, and missed opportunities. We contend that

R.E. Wray (✉) · J.T. Folsom-Kovarik · D. Schmorrow · R.M. Jones · R. Marinier
Soar Technology Inc., 3600 Green Court Suite 600, Ann Arbor, MI 48105, USA
e-mail: wray@soartech.com

J.T. Folsom-Kovarik
e-mail: jeremiah@soartech.com

D. Schmorrow
e-mail: dylan.schmorrow@soartech.com

R.M. Jones
e-mail: rjones@soartech.com

R. Marinier
e-mail: bob.marinier@soartech.com

S. Schatz and M. Hoffman (eds.), *Advances in Cross-Cultural Decision Making*,
Advances in Intelligent Systems and Computing 480,
DOI 10.1007/978-3-319-41636-6_3

this gap derives from differences in representation. Humans tend to represent and understand their world in terms of stories [1, 2], while computer reasoning tends to require formal, mathematical representations.

We are exploring ways in which computers can parse human stories and use collections of stories as a form of causal modeling. The approach, *story-based causal inference* (SBCI) includes both causal inference and analogy to comparable stories. The goal of SCBI is to enable computers to reason about and extend human stories. We hypothesize that bridging this representational gap will allow computers to help humans in new ways. For example, they can assess humans' plans and raise alerts about potential problems, identify potential problems or issues gleaned from reviewing stories about past events, and can support human-computer teaming without introducing technical burden on the human users.

This paper describes an initial proof-of-concept implementation of SBCI. We focus on the domain of military task force planning that requires consideration of political, military, economic, social, infrastructure and information (PMESII) systems. PMESII domains are especially apt for the challenge of story-based reasoning because complex causal interactions within and between these systems are difficult for even experts to understand and predict. In a planning context, where the planners may not have deep expertise in either PMESII systems or direct knowledge and experience in the area of deployment, computer-based support to identify potential interactions and implications may provide immediate impact in helping form better plans and more robustly identify points of potential failure and contingencies. In this PMESII context, we introduce the SBCI prototype, illustrate with some specific examples of the capability it provides, and summarize an initial software test of the prototype.

2 Illustrative Challenge: PMESII Modeling for Planning

Military organizations have invested significant research and resources in the development of processes to make the planning process more routine and less fragile to the limited expertise of individuals within a (multi-echelon) planning team. However, missions and requirements grow more complex and analytical methods are sometimes insufficient for today's complex missions. This is even more evident when a deployment must be planned in a short amount of time, which both limits what options can be considered and the feasibility of bringing in (many) experts to help widen and deepen planners' understanding of the mission in the context of its particular plans.

Short-term planning horizons, limited intelligence, and limited expertise compound the dangers of necessary assumptions in the planning process. Unacknowledged and incorrect assumptions lead to surprises, missed opportunities, and failures to consider second- and third-order effects of plans. Further, one of the most difficult tasks in the planning process is to generate and assess future states, to make meaningful forecasts, even when one's assumptions are inexact and partially informed. Even experts with years of experience often are no better than novices in their ability to assess potential future states [3]. Thus, a core technical challenge of computer-supported planning is to enhance a planner's ability to surface and test assumptions, to help identify important and/or likely possible futures, and to understand how one's assumptions interacts with those possible futures.

Consider, as a specific example, mission planning in a Joint Task Force (JTF). Such task forces are typically faced with complex, novel, and urgent missions. This combination of factors makes planning more difficult and does not provide time for in-depth study and analysis of all factors that may impact the mission.

To illustrate, a JTF might be formed for the purpose of providing disaster relief immediately after a foreign natural disaster such as a typhoon striking a large island city in the Pacific. Such a problem requires the planning team consider many different factors such as the political, economic, military, social, infrastructure, and informational (PMESII) [4] systems within the society. Although good models of these factors may be present prior to the crisis, they most likely will not have been updated to reflect the impact of the crisis on these PMESII factors (washed out roads, disjointed political coordination due to downed communications, etc.). Additionally the presence of an outside military force, even in a humanitarian deployment, can drastically impact any prior models and expectations.

The prototype we describe is exploring how we can capture and use stories about an immediate disaster-relief situation and stories of similar past situations could let a human describe the disaster and their plan for relief, while a computer reasons about the human's stories and responds to them in order to enhance the human's understanding and plans. A desired outcome of this interaction is that the computer system would identify potential gaps, misconceptions, or unstated assumptions in the user's story. By acting on recognized gaps, the system could then assess the humans' understanding and plans, alerting them to important assumptions or opportunities overlooked in the time-stressed planning process.

3 Story Graphs

Human understanding of the world is not simply symbolic but is also grounded in stories. Human experience is subjectively narrative with its cognitive basis in the episodic memory that links memories of events in the context of a linear timeline [5]. Humans understand which plan actions and outcomes are plausible using this narrative experience. As above, we hypothesize that stories can provide a bridge between computers and a human planner.

Computer representations of stories exist [6, 7] and we are drawing on such prior work to inform our representation of stories. At a high-level, a *story* in our conception (Fig. 1) is a sequence of statements about world states and state-changing events, which are represented in a story graph. The story graph allows unification of the representation of knowledge and data from a multitude of sources such as past experience, real-time mission data, computer simulations, and expert knowledge.

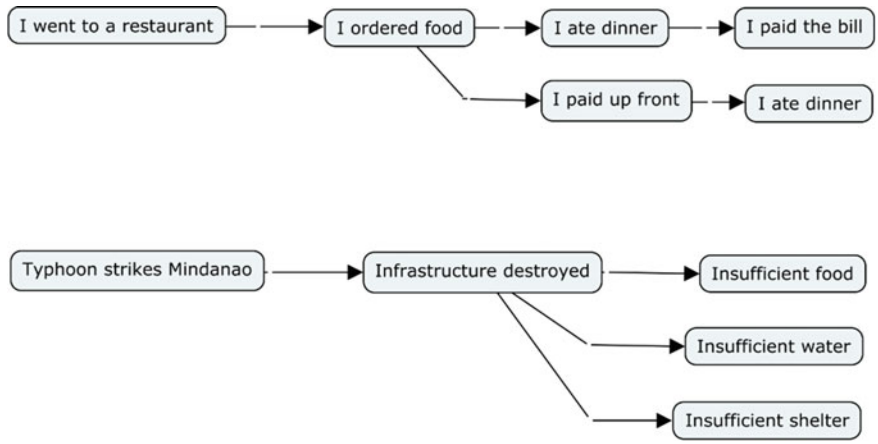


Fig. 1 Examples of simple stories

Nodes in the story graph describe events in the sequence. Links express the concept that a preceding event can cause (or contribute to the cause of) the subsequent event. At this point in our research, we are imposing minimal requirements on the semantics of nodes and links to provide greater flexibility in exploring how to use these stories for computer-based inference. For example, we expect to adopt in the future a richer vocabulary of linkages, so that graphs can express AND, exclusive OR, relative contributions, and so forth. Other researchers have described formal definitions of the kinds of causal links that can appear in stories [8]. However, there is no agreed standard so we took a least-commitment approach in defining causal linkages in this initial research.

Conceptually, stories are recursive so that a high-level node ("I ordered food") could be composed into a more detailed representation of the event ("I asked about the specials", "The server said…"). Eventually, nodes (and thus stories) are grounded in semantic knowledge of the world that makes each described event and state have meaning in the whole [9]. These story graphs can relate what could plausibly happen next, as opposed to what must deterministically happen. The graph structure allows for branching and merging, representing alternative linear paths, of which more than one may be true.

4 Story-Based Causal Induction

Figure 2 illustrates, at an architecture level, how the prototype system we have developed uses story graphs to generate suggestions for consideration during planning. Although our research program encompasses methods for extracting computer-based stories from other sources, such as natural language (stories from a

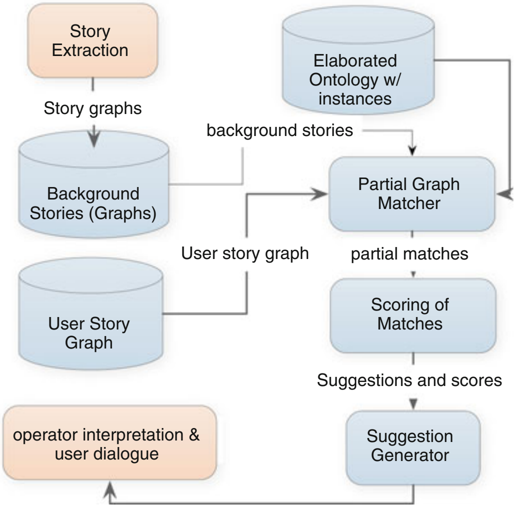


Fig. 2 Conceptual view of analogical story matching

newspaper), sensors, non-text sources (e.g., maps) and from users directly, detailed functional representations of these systems are elided in this view (orange boxes).

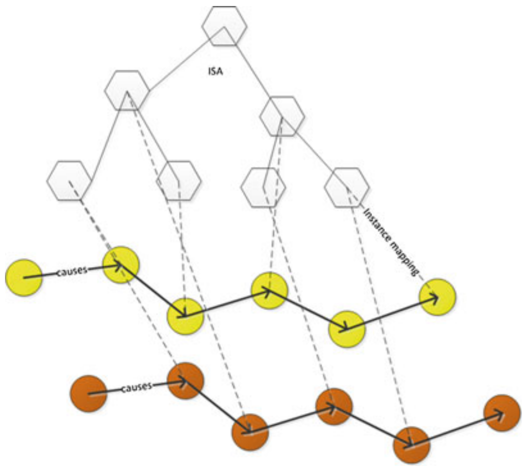
The primary focus in this paper is to describe how we can use past stories to inform or enrich a developing plan (where a partial plan is an "in progress" user story). We term this process story-based causal inference (SBCI), which is represented in the figure with the blue boxes. Story-based causal inference allows the computer to reason about, connect, and integrate story graphs. By finding relevant knowledge in a story graph knowledge base and combining them with the story of the current situation, the computer system can help human users surface unstated assumptions and identify unexpected outcomes, improving their foresight and mission effectiveness.

SCBI employs a common story graph representation, whether individual stories derive from users or background stories. They can be elaborated with semantic knowledge from an ontology. User stories are matched to other stories via a partial matching process. The SCBI prototype uses partial matching for two complementary reasoning methods, analogical matching and causal inference, which we describe further below. Resulting partial matches are then scored and the highest scoring matches are translated from story graphs to a more human-understandable text representation for presentation to users.

4.1 Analogical Matching

Analogy [10, 11] is apt for finding possibly relevant information and incorporating it into a story graph to elaborate or extend user stories. Analogy is also familiar to military decision makers who are expected to create mission analyses that draw on

Fig. 3 Conceptual view of analogical story matching



knowledge from past events with similar characteristics or in the same geographical region. We drew on cognitively inspired planning algorithms that carry out goal-based graph search. They incorporate various heuristics designed to overcome worst-case computational complexity of graph matching by leveraging domain knowledge and machine learning [12]. We drew on elements of this prior work to create a prototype that could draw analogical connections between different but similar story graphs.

The computational problem is represented conceptually in Fig. 3. A user story (orange) and a prior story (yellow) have similar structure. Representational elements within a node are mapped to a common ontology (hexagons) as instances. For example, in the prior story illustrated in Fig. 4, the first node would be mapped to the political entity *Haiti* in the ontology and the natural disaster *Earthquake*. These mappings to the ontology then enable additional connections via ontological relationships. The ontological representation of Haiti might link further to its political and military organization, its economic structures, and so on.

Matching of nodes in stories can be exact or partial. Exact matches correspond to finding a previous occurrence of the same conditions in a past story. For example, a common node appearing in many disaster relief stories might be *lack of potable water*. If the same node appears in the user story, the system could immediately match it to many past stories. In Fig. 3, the second node in both the user story and

Fig. 4 An example historical story used in analogical mapping

the past story are mapped to the same node in the ontology, representing an exact match.

Most graph matches are not exact. Partial graph matching uses the ontology to help the analogical matching process identify possibly relevant or interesting matches. With these augmentations, the prototype can carry out different types of analogy:

1. **Generalization** makes use of the ontology links to attempt matches between nodes that talk about different objects, but the objects share a common ancestor in the ontology (consider the third nodes in the user and past story). For example, a generalization of the current situation, which contains a story node medical service degraded, might infer that the node should match to a past story with the node *police service degraded* because both objects are subclasses of services.
2. **Transformational analogy** [13] refers to possible analogies between relations that the system can hypothesize might exist and uses graph structure rather than a match on node contents. For example, if a node contains the relation *is-present-in(disease, region)* then the system might recognize a structural alignment with another node that contains *occurred-in(disaster, region)*. Structural similarity can be enhanced if additional nodes and edges match across the two graphs.

Completing the analogy requires inference from a matched past story to the current situation. In the prototype, we remap objects within nodes from the current situation story to the matched graph. For example, if the current situation contains *disease is present in Affected Region* and the analogy is to *disease in a region degrades medical services*, then the remapped inference produces *this medical service in Affected Region will be degraded*. This subgraph appears as a new node dependent on the original node with a linking edge. In the prototype, all inferences led to adding nodes in a story. The new node also becomes available for iterative application of the matching algorithm to find more suggestions that follow from an inference.

4.2 Causal Inference

In some cases, users of a system such we are developing may want to simply assert that some causal relationship exists, rather than needing to derive or discover one thru analogy. The example in Fig. 5 presents an inference rule that says that disease conditions are more likely to increase in effect in a region when the medical service in that area has been degraded. The nodes preceded by "=" serve as variables, so that the rule is applied to a particular medical service, region, and disease. Such a rule could be used, for example, to anticipate specific disease outbreaks following degradation of medical services in an area after a typhoon.

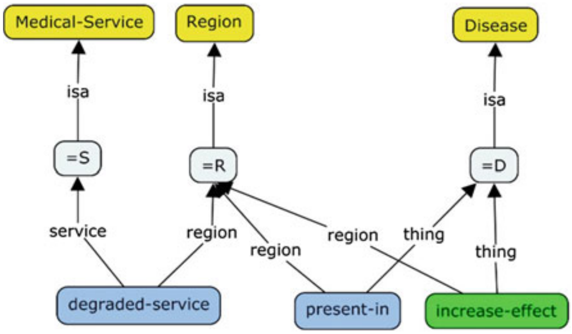


Fig. 5 An example graph-based inference rule

From an information content perspective, these rules capture and express causal properties that may not be evident in individual stories. They can fill in gaps and help the system be more efficient in finding relationships and connections. Computationally, they are expressed in the prototype in the same story graph representation that is used to capture user stories and past histories. The ontological mapping process outlined earlier allows the variables defined in the rule to be bound to specific elements from user stories.

This design choice has one immediate benefit and one potential long-term benefit. The immediate benefit is that the prototype can use the analogical mapping process presented earlier to apply these causal rules. The same underlying computational process is used for both analogy and causal inference. The long-term benefit of this uniformity in representation is that it can support the gradual transformation of analogical connections between stories to causal rules, enabling a kind of causal learning. For example, after the observation of many typhoons, hurricanes, earthquakes and other natural disasters and the outbreak of different diseases following these disasters, the type of system we are developing could likely learn the kind of rule represented in Fig. 5.

4.3 Match Scoring and User Feedback

Many possible matches could be generated by the partial mapping outlined above. The prototype includes a scoring function that attempts to estimate the logical sufficiency and logical necessity of individual graph components with the goal of assessing the "aptness" of the potential inference. For example, if a subgraph of a user story is also present in a past story, then the prototype estimates the logical sufficiency (LS) of the nodes in the augmented subgraph. If a subgraph of the user story is missing from a candidate past story, then the system estimates the logical necessity (LN) of the missing nodes. In other words, each matching node improves the match score and each missing node decreases the match score. The scoring function was tuned manually for the software test event. Long-term, it would be preferable for the system to learn the relative weights of necessity and sufficiency matches.

Scoring of candidate matches are currently used to rank the candidates for presentation to the end user. The same scores could be used to filter and prioritize the alerts that result from story additions. However, the current scoring only takes into account the "goodness" of the match. We expect that users will also be interested in seeing story elaborations that are relevant or important for their plan. This expectation was confirmed in the system test event: match scores by themselves did not contain all the information needed to determine which story additions should result in alerts. There is also a need to estimate the likelihood and the impact of each story addition in order to highlight alerts that make a difference and to justify the suggestion to an end user. Story suggestions should be relevant to the users plan. To-date, we have focused on two types of suggestions: surfacing an unstated assumption and identifying an unexpected possible outcome of the situation.

1. **Surfacing an unstated assumption** is represented by the insertion of a story node that has a directed edge pointing into an existing node. For example, a user might create a story graph about disaster relief that contains the two nodes *lack of potable water* → *water delivery*. By analogy to previous stories, the system might have access to multiple nodes that in the past have been linked to *deliver water* (e.g., *truck convoy*, *set up desalination*). The system could then suggest any of these as possible assumptions that might be underlying the user's story. If more information is available from the rest of the user story or from any mission parameter inputs, the system could use the information to narrow the choices. For example, a desalination plant resource might suggest that the unstated assumption in water delivery was that the desalination plant is available.
2. **Identifying unexpected outcomes** is represented by the suggestion of a story node that only has edges pointing into it from the established story. The new node extends the story graph as a whole. Imagine that a past story contains knowledge about the available desalination plant—its capacity is only designed to support the personnel embarked on its own ship, so supporting more people in the stricken population will lead to reduced water available to the rescuers. (Such knowledge could come from a narrative when the desalination plant was overburdened, or more simply from general expert knowledge about the tool.) If the system suggests a new node with the content *not enough capacity*, the new node is a result of the existing node *set up desalination* and is thus an unexpected outcome. The combination of the two nodes together, a surfaced assumption and an unexpected outcome that invalidates the assumption, together show how the prototype could raise a salient alert to the end user.

5 Prototype Test and Results

We conducted a small-scale software test of story matching and inference. Five testers (subject matter experts with experience in military planning) interacted with the system individually over the course of two hours. In the first hour, they created stories that reflected their analyses of a disaster relief scenario.

Operators from the research team translated these stories directly into story graphs because our goal was to test the functionality of SBCI, rather than the adequacy of user interfaces for capturing user stories. The stories the testers created were then used to test SBCI in the second hour. We used a background corpus of natural disaster events (e.g., Hurricane Katrina) and location-specific data (similar to *CIA World Factbook*), encoded into story graphs (2,575 total story nodes connected to an ontology of about 9,000 classes). We then applied SBCI to the planner's stories and this background knowledge to attempt to generate examples of unsurfaced assumptions and unexpected outcomes in the testers original stories.

Results of the system test showed that the testers were able to create story graphs describing their analysis. The captured graphs contained on average 105 nodes (67–133). The system then offered an average of 16 story suggestions (11–36) for each tester. The testers accepted on average 6 suggestions (2–11). Table 1 summarizes a few examples of suggestions generated by the prototype that were accepted by the testers and added to their stories.

Even when they did not accept a change, the testers sometimes did make other changes that were related, so the suggestion may have served as a reminder. We also observed that testers applied some interpretation to accepted suggestions. For example, the elaboration in Table 1 about the potential for violence resulted in an

Table 1 Examples of story elaborations produced by SBCI in the software test event

Type	Original plan	Elaboration
Unstated assumption	Restore unlading capability and bring in supplies and building materials thru the seaport	Storm effects, in addition to lading deficits, may limit navigational aiding and docking capacity
Unstated assumption	NGO can facilitate delivery of medical services in the absence/degradation of local medical services	NGO offers technical expertise and capacity in other areas, including water supply and communication with population
Unstated Assumption	Establish communications with provincial government	Local government includes an autonomous region with a distinct governance structure
Unexpected outcome	Establish food distribution points in rural areas	Local government may actively hinder food distribution to areas that are "troublesome"
Unexpected outcome	Establish food distribution points in urban areas	Fighting and violence sometimes occur at food distribution points in urban areas following disasters

elaboration of the planner's story to assess security needs rather than to expect violence.

One of the limitations of the test was the relatively small amount of time allowed for a large planning problem. As a result, many suggestions may likely have been evident to planners given more time. Although they were free to draw on their own knowledge, the planners did have available (during both hours of the event) all the source material that was encoded as background stories; that is, all the data that the system used to make its suggestions was available to the testers. This represents the practical constraint that sometimes available knowledge cannot be retrieved or applied in time-constrained contexts, in both human and computational systems [14].

The test event also highlighted several complementary use cases for story-based causal inference. For example, in a JTF mission analysis and planning scenario such as we tested, it might be necessary for users from multiple echelons (ranks, roles, and expertise) to contribute to a single story. This use case highlights the need for a hierarchical story graph, which would enable different users to review and fill in the story with different levels of detail. As a second use case, which we did not focus on in our prototype but test users expressed a desire for, it should be possible for the story representation to identify a "hinge" (crucial decision point) or pattern in mission data. For example the story should be able to identify that one possible branch is likely to lead to the same completion under a number of initial conditions, while another branch has good possible outcomes but also several negative outcomes are possible.

6 Conclusions

Our long-term goal is to explore the impact of using stories to bridge the gap between human and computer understanding. We hypothesize mutually shared representations will improve human-computer teaming and mitigate the common and sometimes disastrous mistakes that arise from misunderstanding or misapplication of information by computers or humans by the other.

The story-based causal inference prototype offers positive but preliminary evidence that a shared, story-based representation can be used constructively and collaboratively by humans and computers. In the context of the military planning example, it allowed unstated assumptions to be surfaced and unexpected outcomes or implications to be identified during planning. It drew on data about local PMESII systems and prior stories of natural disasters to help deepen and extend testers' available knowledge of their plans, such as the need to consider two parallel governmental authorities in one region, or the socio-economic factors influencing food distribution in different areas. These suggestions would not transform a typical military planner into a PMESII expert, but they would aid planners unfamiliar with the intricacies of PMESII systems to better factor PMESII considerations into plans.

Although promising, additional work is needed to make SBCI practical for the military planning use cases described in the paper. We see three core areas of future research and development:

1. **Scalability**: SBCI employs heuristics to limit the impact of the costs of general graph matching, but its actual scalability to real-world problems is not yet established. We expect gains in scalability will come from a richer vocabulary of causal linkages. Different kinds of links and indices to them will help the partial matching process target subgraphs within the large corpus of background stories more efficiently. However, this direction is in tension with the relative simplicity and understandability of the linkages to potential users. Thus, the goal is to find a representation that remains readily understandable, while providing sufficient discriminating power for matching.
2. **Semantic Integration**: The prototype used a standard ontology and some general mapping rules to connect elements within story nodes to the ontology. For the future, a more sophisticated mapping is needed, along with both larger and more domain-specific ontologies. Better mappings will enable improved partial matching and more targeted match scoring. For example, analogical match could be made more precise if the matching process could traverse the ontology along with the story in deciding a potential match. In the prototype, we used only ISA mappings. For match scoring, a more domain-specific ontology would facilitate scoring based on relevance in addition to semantic and structural similarity.
3. **Learning and Generalization**: The current prototype does not improve its performance with experience. Several different kinds of learning would improve overall performance. Most importantly, as discussed above, a process that allows the system to learn new causal inference rules would both help improve capability and scalability because causal rules rely more on exact match than partial match.

Acknowledgments This work is supported by the Defense Advanced Research Projects Agency project HR0011-15-C-0067. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of Defense or the Defense Advanced Research Projects Agency. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. We would like to thank collaborators and sponsors at DARPA who provided insights and operational perspectives in the development of story-based causal inference and Mike Lerario and Kevin McEnery for their insights and perspectives on the challenges of military planning.

References

1. Kahneman, D., Tversky., A.: The simulation heuristic. In: Kahneman, D., Slovic, P., Tversky, A. (eds.) Judgment under Uncertainty: Heuristics and Biases, pp. 201–208. Cambridge University Press, Cambridge UK (1982)
2. Fisher, W.R.: Human Communication as Narration: Toward a Philosophy of Reason, Value, and Action. University of South Carolina Press, Columbia, SC (1989)
3. Tetlock, P.E., Gardner, D.: Superforecasting: The Art and Science of Prediction. Crown Publishers, New York (2015)
4. Zacharias, G.L., MacMillan, J., Susan B. Van Hemel (eds.): Behavioral Modeling and Simulation: From Individuals to Societies. National Academies Press, Washington, DC (2008)
5. Tulving, E.: Elements of Episodic Memory. Oxford University Press, Oxford (1985)
6. Schank, R., Abelson, R.: Scripts, Plans, Goals, and Understanding. Erlbaum, Hillsdale, NJ (1977)
7. Richards, W., Finlayson, M.A., Winston, P.H.: Advancing Computational Models of Narrative. MIT CSAIL (2009)
8. Shapiro, B.P., van den Broek, P., Fletcher, C.R.: Using Story Based Causal Diagrams to Analyze Disagreements About Complex Events. Discourse Processes (1995)
9. Verheij, B.: Virtual Arguments: On the Design of Argument Assistants for Lawyers and Other Arguers. TMC Asser Press, The Hague, The Netherlands (2005)
10. Hofstadter, D.R.: Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought. Basic books (2008)
11. Gentner, D., Holyoak, K., Kokinov, B.: Analogy: Perspectives from Cognitive Science. MIT Press, Cambridge, MA (2000)
12. Jones, R.M.: Problem solving via analogical retrieval and analogical search control. In: Foundations of Knowledge Acquisition, pp. 227–262. Springer (1993)
13. Kuchibatla, V., Muñoz-Avila, H.: An analysis on transformational analogy: general framework and complexity. In: Advances in Case-Based Reasoning, pp. 458–473. Springer (2006)
14. Newell, A.: The knowledge level. Artif. Intell. **18**, 82–127 (1982)