Article

Studying Consensus and Disagreement during Problem Solving in Teams through Learning and Response Generation Agents Model

Alex Doboli [1] **and Daniel-Ioan Curiac** [2,*]

[1] Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794-2350, USA; alex.doboli@stonybrook.edu
[2] Department of Automation and Applied Informatics, Politehnica University of Timisoara, V. Parvan 2, 300223 Timisoara, Romania
* Correspondence: daniel.curiac@aut.upt.ro

Abstract: Understanding the process of reaching consensus or disagreement between the members of a team is critical in many situations. Consensus and disagreement can refer to various aspects, such as requirements that are collectively perceived to be important, shared goals, and solutions that are jointly considered to be realistic and effective. Getting insight on how the end result of the interaction process is influenced by parameters such as the similarity of the participants' experience and behavior (e.g., their available concepts, the produced responses and their utility, the preferred response generation method, and so on) is important for optimizing team performance and for devising novel applications, i.e., systems for tutoring or self-improvement and smart human computer interfaces. However, understanding the process of reaching consensus or disagreement in teams raises a number of challenges as participants interact with each other through verbal communications that express new ideas created based on their experience, goals, and input from other participants. Social and emotional cues during interaction are important too. This paper presents a new model, called Learning and Response Generating Agents, for studying the interaction process during problem solving in small teams. As compared to similar work, the model, grounded in work in psychology and sociology, studies consensus and disagreement formation when agents interact with each other through symbolic, dynamically-produced responses with clauses of different types, ambiguity, multiple abstraction levels, and associated emotional intensity and utility.

Keywords: agent models; problem solving; symbolic representations; optimized teams

MSC: 68T42; 93A16

Citation: Doboli, A.; Curiac, D.-I. Studying Consensus and Disagreement during Problem Solving in Teams through Learning and Response Generation Agents Model. *Mathematics* **2023**, *11*, 2602. https://doi.org/10.3390/math11122602

Academic Editor: Florin Leon

Received: 4 May 2023
Revised: 3 June 2023
Accepted: 5 June 2023
Published: 7 June 2023

1. Introduction

The core of many team-based applications is the process of reaching a set of commonly agreed-upon ideas [1]. For example, problem framing identifies, prioritizes, restructures, and distills the requirements of a problem [2,3]. Problem framing might require finding creative solutions to improve the healthcare system in the USA [4] or making learning in a university setting more fun [3]. It must find the requirements that are collectively perceived by a group as important and useful [5–7]. Similarly, goal identification searches for the common goals that inspire and drive a group and thus have value (utility) for the participants [8]. Problem solving identifies solutions and solving methods considered realistic and effective by the group [9]. Other related applications include optimizing team performance through adjusting team composition, interaction, and behavior [10], or devising training procedures to improve the group activity of certain individuals [11]. Human computer interfaces (HCIs) can also benefit from studying this process, as creating a digital twin representation [12] of a user can help self-improvement by showing opportunities to improve education [13], comfort [14], and growth [8]. For example, ref. [13] proposes a system that tracks coding and debugging to create a digital twin of a student's learning programming so that any learning shortcomings are identified and then addressed through customized interventions.

The above cases refer to participants that interact by stating their ideas generated in response to others' inputs and their own beliefs and experiences. Emotional and social cues are important in the process too [15,16]. The exchanged ideas are syllogisms [17]. More precisely, ideas are "packets of meaning" [18], e.g., structures in which concepts (i.e., nouns) have a role within a structure and are connected to each other by verbs. Concepts and verbs are qualified through attributes and adverbs, respectively. A response is consistent with the experience of the generating participant and, similarly, is understood by another agent within the context of its own experience. Finally, the purpose of interaction is not to optimize parameters of statistical models (like discussed in the related work section), but to create outcomes (e.g., responses) that are either commonly agreed upom by the participants (consensus or entrainment) [1,19] or represent a segregation of the participants based on their beliefs, priorities, and experience.

Various kinds of ambiguities must be addressed by participants during their interaction process. Each response is understood by a participant based on his/her experience, the produced outcomes, and the representativeness of the outcome for the agent. A first kind of ambiguity originates due to the specific experience of each participant, which results in different understandings of the same response. A participant's understanding of a response is always slightly different from that of another participant or from the intended meaning of the originator. A second kind of ambiguity emerges due to the gap between the goals and needs of a member and the solutions found to address them. Arguably, a solution can never fully address each member's goals and objectives. Finally, the third kind of ambiguity arises from the outcomes found for a goal and the goal itself. The nature and size of the gap between goals and outcomes relate to the participant's incentives and motivation to participate in the interaction process.

In summary, problem solving in a team produces a sequence (trace) of responses that try to find the best consensus (agreement) between the produced responses and the goals, objectives, and preferences of the team members. It has been argued that a trace describes the high-level cognitive processes of the members [19]. As members have goals, beliefs, and experiences that do not change during problem solving, the consensus found must be valid under multiple assessment references. Responses are structures of clauses with defined or partially defined roles, so that the response is consistent with the generating member's experience. Three kinds of ambiguities can occur during response understanding. Finally, responses are created using procedures that combine clauses from experience and received responses under the influence of emotional and social cues. These issues have rarely been addressed by existing team-oriented models. Current models mainly consider interacting agents that execute predefined actions on stochastic data guided by a cost function but study less learning and the generation of symbolic knowledge. Most models do not consider emotional or social cues either.

This paper proposes a novel agent-based model to study team-based problem solving that results either in a consensus among members if they agree on solutions, or in dividing members along their favorite solutions if there is disagreement between them. The goal of the parametric model is to get insight into how parameters influence the process that leads to either consensus or division. The two states can express various aspects, such as needs in the case of problem-framing applications, goals in the case of finding shared utilities, or solutions and solving methods. A consensus or division is reached if there are no significant modifications to responses and if there is agreement on their associated utility. Model parameters refer to the main primitives of the model, such as characterizing the similarity among concepts (e.g., similarity of their meanings), associations between concepts and their utility, structural frames used to create responses, addressing ambiguities in responses, and combining responses to produce new outcomes.

The proposed model is based on Learning and Response Generating Agents (LRGAs) that interact with each other through responses, which are sentences (structures) with nouns, verbs, adjectives, and adverbs. Knowledge (e.g., inputs, experience items in the agent's memory, and responses) are symbolic networks of concepts connected through relations describing a role in the expected outcome (as depicted in Figure 1b). An LRGA learns by updating its memory and concept access frequencies based on received ideas and its own responses. As shown in Figure 1a, each LRGA first understands a received input by finding the best match and difference between the input and the statements in the agent's memory, similar to the alignment process in psychology [20]. Word meaning is defined using state-of-the-art databases in linguistics, such as the WordNet database [21]. The role of some of the response components (called clauses) can be ambiguous. Section 3.2 presents the input understanding method. After input understanding, an LRGA creates new responses by combining concepts through strategies selected based on utility differences and emotional and social cues. The strategies used by agents are enumerated at the bottom of Figure 6. These strategies can perform the following activities: explore variants of a response by adding more details or considering alternatives with similar meaning, establish abstractions for a set of similar solutions, pick up new, less frequent concepts, and reason with more abstract solutions by combining their related sentences. An LRGA's motivation to participate in the response generation process is determined by the difference between an expected outcome and a received response. This difference is usually considered the source of various emotions. Experiments discuss agent parameters that favor agreement or disagreement between agents. The LRGA model was devised to study problems that require generating new ideas accepted by all team members, such as problem-framing and common ground identification, and to support new HCI for self-improvement and teaching.

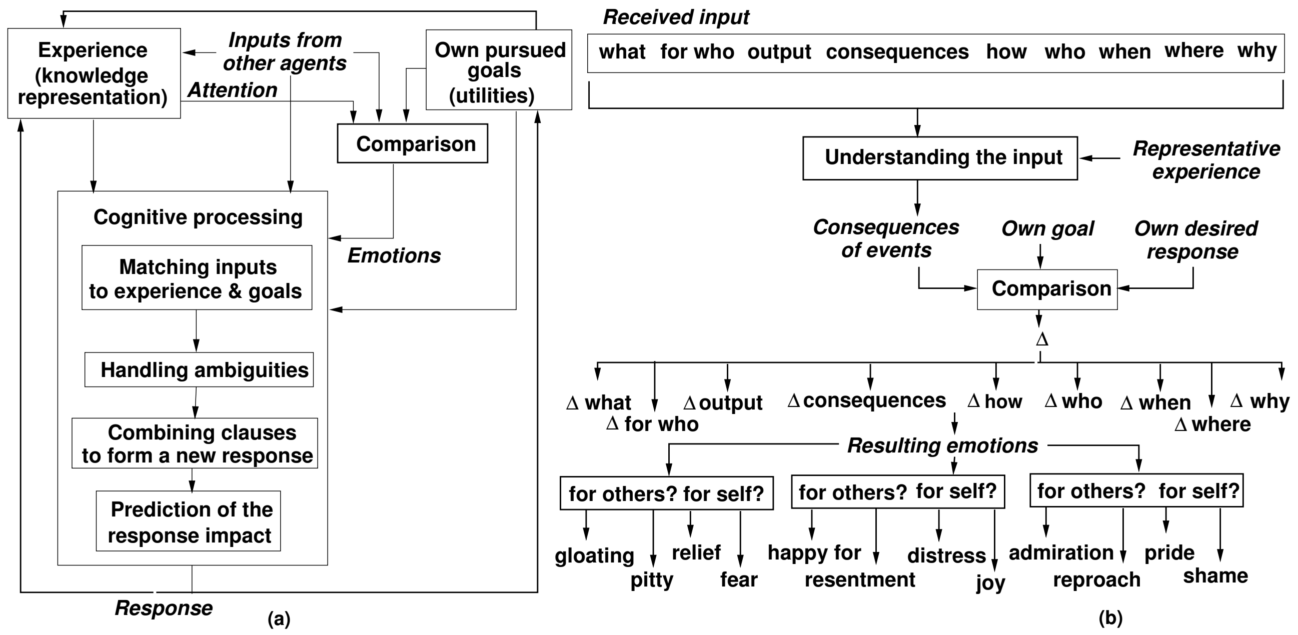

Figure 1. (**a**) Cognitive processing to generate a new response, and (**b**) matching and comparing two ideas.

As detailed in Section 2, this work brings the following contributions as compared to related work: (1) Grounded in theories in psychology and sociology, the model focuses on understanding how agent interactions in small teams result in consensus or division. (2) The model considers verbal structures as the main interaction element, which is a more accurate expression of human communication [17]. Its semantics include clauses of different types, word ambiguity, multiple levels of abstraction, and social and emotional aspects. (3) Agents create new responses based on their goals and evolving experiences, in which social and emotional cues are important.

The paper has the following structure. Section 2 summarizes related work. Section 3 presents an overview of the LRGA model, followed by response modeling, ambiguity solving, and response generation. Section 4 presents the model implementation. Experiments are presented in Section 5. Section 6 discusses the model as compared to related work. Section 7 end the paper.

2. Related Work

Psychology and sociology have approached team-based problem solving as a cognitive process conducted in a social context [1]. In this approach, joint action represents the interaction among team members that aim to achieve common goals, intentions, or ground [1,8,22]. Interactions coordinate members through synchronization, entrainment, alignment, and convergence [6,10] and depend on the conversational context and the team members' intentions and features [23]. For example, affiliative conversations have different interaction characteristics than argumentative interactions. Abney et al. explain that coordination is a complex process that occurs at different time-scales and hierarchical levels [10]. A complex system of behavior emerges because of explicit and implicit matching at the phonetic, phonologic, lexical, syntactic, semantic, and situational levels. However, member coordination goes beyond speech attributes. Common ground knowledge, shared visual information, and beliefs about the other team members influence postural and gaze coordination [19]. Affective valuation is also important during interaction [16,24], such as different interpretations of the external stimuli and distinct expectations for the outputs of problem solving. Members must be committed to participating in the team effort.

Related models in psychology and sociology propose different scenarios for interactions in a team [1]. "Visual worlds", arguably the simplest model, assumes that team members follow an active-passive participant approach with no changing initiative and little coordination between members [25]. The message model states that communication is a probabilistic information flow at a certain rate, during which the sender and receiver must employ the same encoding and decoding of the "packets of meaning" represented by words [18]. Social interactions are less important, as there is no coordination, intention recognition, role taking, or matching between the communicated details and the shared common perspective [1]. In contrast, social aspects are addressed in the two-stage model that focuses on lexical entrainment, shared perspective, and reuse of syntactic forms. Fowler et al. indicate that parsing and speaking durations, speaking rates, turn durations, response latencies, vocal intensities, and accents depend on coordination [26]. The interactive alignment model considers members' perspective adjustment and the creation of mental models about other team members [27,28]. Lowe et al. propose an extension to the Associative Two-Process (ATP) theory to include social cues and affective states [24]. The model uses temporal difference learning to express expectations and to include social cues. Learning considers the magnitude and omission of rewards as well as the temporal difference between stimulus and learned outcome. The work argues not only for the importance of a member's intentional behavior towards a goal but also for the need for an agent to learn the other's behavior and then adjust accordingly. Finally, grounding models emphasize the social, collaborative view, i.e., coordination of meaning, observation of each other, and creation of mental models about others [1,19].

Computational models for describing team interaction have been proposed by work on Multi-Agent Systems (MAS) [29–33]. KABOOM is the first agent model devised to study the impact of the agent's cognitive style on problem solving [34]. Based on the Kirton Adaptation-Innovation Inventory (KAI), agents can prefer incremental adaptations or radical changes to a solution. The model was used to solve engineering problems, such as race car or beam design problems [35]. A MAS model studies the spreading of cultural traits depending on the agent's feature similarities [36]. However, most MAS models are not grounded in psychology or sociology but instead attempt to solve complex distributed optimization problems such as search and resource management in smart traffic systems, cooperative navigation, UAV driving, and IoT communication [37–39].

Agents explore different or partially overlapping problem sub-spaces by using their own experience, objectives, and interpretation of the current situation. Some agents implement Decentralized Partially Observable Markov Decision Processes, in which a centralized critic is in charge of learning while agents operate decentralized [37,40]. Learning maximizes the gradient of the next expected return or sequence of returns. A separate critic is proposed for each agent in ref. [37], but at the cost of a lower learning convergence [40]. MAS models also use dynamic learning to optimize the networked connectivity between agents, e.g., network topology and parameters, and control policies [37,39,40]. An important problem is deciding the communication targets, e.g., sending a message only to the agents that need it [40]. A solution is to use soft attention represented as weights depending on the softmax criterion of the similarity of a message's signature and a query vector describing an agent's hidden state [40]. The state of the common critic comprises all agent states and is updated using all messages between agents. An alternative is presented in ref. [39], in which messages of limited variance are sent to the receiving agents. Each agent locally generates actions using a Gated Recurrent Unit (GRU) that creates an intermediate value using the agent's state and observation, followed by a Fully Connected layer that encodes the intermediate value to generate the associated action value. Each agent combines its own action values with those produced by the encoders for the messages from each other agent. Another solution describes attention by comparing the embedding through a Multi Layer Perceptron of an agent's observation and actions with the embeddings received from other agents [37]. A two-level attention model is proposed in ref. [38]: hard attention describes the agents to which one must pay attention, and soft attention presents the weight of the attention.

A distinct body of work on MAS includes models in which agents communicate through commitment-based protocols [30,41,42]. A commitment protocol assumes that agents interact by assuming the debtor or creditor role, which implies agents' commitment to comply with the protocol. A social commitment is controlled by an antecedent and produces a consequence. A creditor agent makes its own decision based on its goals and beliefs to detach from a social commitment. Each commitment protocol is a sequence of actions, such as create, select, cancel, release, detach, satisfy, discharge, expire, violate, assign, and delegate, that pursue a well-defined life-cycle, including states such as violated, satisfied, and expired [30]. The formal definition of commitment antecedents and consequences as logic propositions as well as the predefined sequence of actions of each protocol supports protocol synthesis [43] and proving properties about agent teams [42]. As compared to message-based agent interactions [44], which prescribe an automaton-like sequence of steps, commitment protocols are more flexible in describing the nature and parameters of an agent's interactions [30].

The learning and response generation agent (LRGA) model in this paper does not target distributed optimization or guaranteeing and proving properties about interacting agents, like related MAS, but instead focuses on getting insight on how the parameters of agent interactions influence open-ended problem solving in small teams. The LRGA model is grounded in theories in psychology and sociology, such as the joint action interaction process. It arguably includes more complex semantics and social and emotional descriptions. Instead of logic or statistical descriptions of knowledge and reaction, the model considers verbal statements, which are a more accurate expression of human interaction [17]. Furthermore, inputs, experiences, and responses are dynamic symbolic networks of concepts at different levels of abstraction, defined by representative exemplars, and connected through the relations defined by verbs. Network structures change based on the context-specific understanding of responses. New responses are not the result of Markovian Decision Processes as the states of an agent can change based on the joint interaction process. Instead, agents respond based on their goals and evolving experience, in which social and emotional cues are important. Moreover, convergence does not simply describe stability, such as in the related work, but rather reaching a state in which every agent feels that it produces the "right" responses according to its experience, goals, and other agents' reactions. The computational complexity due to the distributed nature of teams is low, as the considered teams are small, with about four to five members. Instead, the high complexity arguably relates to the underlying semantics (e.g., responses) of the agent's experiences, goals, responses, and their interpretations. Also, understanding the facets of agent adaptation (adjustment) is challenging, including the importance of shared knowledge and social and emotional interactions. These aspects are important not only for improving problem solving in teams, but in general for any human-in-the-loop Cyber-Social computing application. More information on how the LRGA model was developed based on insights from the related research literature can be found in the following section (Section 3) of the paper.

3. Learning and Response Generating Agent Model Overview

Learning and response generating agents (LRGA) communicate with each other through statements to present their ideas as part of problem solving. LRGAs produce new statements (responses) by a cognitive-processing-inspired method using their own experience, their own pursued goals and priorities (modeled as utilities), responses from other agents, and emotions. Figure 1a summarizes the process through which LRGAs create new responses during their interactions. Responses can be created through two flows [6,45–47]: a fast, top-down flow and a slow, bottom-up flow. The fast flow reacts to the received inputs, present emotions, social interactions, and drawn attention to create responses that incrementally change the input or experience. The slow flow offers planned solution building that uses the agent's learning after generating a number of fast responses. Its responses are at higher levels of abstraction. Urgency selects between the two flows.

Each LRGA executes two main activities as part of agent interaction, input understanding and response creation. Input understanding connects to an LRGA's attention and emotion, which are subsequently used in response creation.

Input understanding establishes a connection between a received input and the agent's own experience (e.g., representations in the agent's memory). In addition to finding similarities and differences between the input and experience, understanding handles any ambiguities that exist after matching, e.g., due to the multiple roles that some input fragments can have. Also, it processes inconsistencies between input and its own experience and goals. Examples in Section 3.2 illustrate ambiguities during understanding. The understanding process is shown in Figure 1b. It has two steps: (a) matching the fragments and (b) stitching (relating) the fragments into the most likely sequence. Matching identifies similarities between the fragments of the input and the fragments of the agent's own experience. Stitching ties fragments together in a new description (i.e., idea), either to indicate a justification (purpose) or to show a consequence between fragments.

After input understanding, an *LRGA's attention and emotions* are computationally modeled by comparing its own experience and goals to the input. The model is based on appraisal theory in psychology [48,49]. Figure 1b illustrates an LRGA's emotion formation. The consequence of an understood input, which is a prediction of the input effects, is compared with the goals of the agent or its previous responses. The comparison produces a set of differences (Δs), with a specific Δ for each input fragment (clause), e.g., *what*, *for who*, *output*, and so on. As shown in Figure 1b, specific emotions emerge depending on the nature and degree of unexpectedness of Δs, and whether Δs refer to the agent (*for self?*) or to other agents (*for others?*). Section 4 details the implementation of the attention and emotion models.

Response generation dynamically combines the accessed fragments into a new response and then predicts the impact of the response, e.g., its utility. A new response is created using the strategies enumerated at the bottom of Figure 6. Based on our previous work in refs. [6,50,51], these strategies explore variants of a response by adding more details or considering alternatives of similar meaning, establish abstractions for a set of similar responses, pick up less frequent concepts, and reason with more abstract responses by combining their structures. More details about the strategies are offered in Section 4, and their implementation is discussed in Section 5.

The remainder of the section discusses the theory behind response structure, input understanding, and response creation.

3.1. Response Structure and Symbolic Representation

The LRGA model assumes that in the most general case, an agent's response relates an objective to an implementation, which is a meaningful way of achieving the objective within a certain purpose [6]. However, in the current software implementation, a response includes only one of the two parts: the objective or the implementation serving a given purpose.

$$objective \leftrightarrow implementation \mid purpose \tag{1}$$

Symbol $\leftrightarrow$ expresses the association between objective and implementation, and a symbol $\mid$ describes the purpose within which the meaningful association exists, e.g., the association has a meaning within this purpose.

An objective is a composition of fragments (clauses) with the roles of *what*, *for who*, *output*, and *consequence*:

$$objective \equiv what \circ consequence \circ (output^{(1)} \circ (for\ who^{(1_1)} \ldots \circ for\ who^{(i_1)})) \ldots \circ (output^{(k)} \circ (for\ who^{(1_k)} \ldots \circ for\ who^{(i_k)})) \tag{2}$$

A fragment *what* declares the nature of the objective, fragment *consequence* introduces the utility of the fragment *what*, fragment *output* expresses an output that refers to the objective, and a fragment *for who* describes the beneficiary of an output (i.e., for whom is the output meant). Symbol $\circ$ describes the composition of the fragments.

An implementation is a composition of fragments (clauses) with the roles of *how*, *who*, *when*, and *where*:

$$\begin{aligned} &implementation \equiv \\ &\quad how^{(1)} \circ ((who^{(1_1)} \ldots \circ who^{(m_1)}) \circ (where^{(1_1)} \ldots \circ where^{(r_1)}) \circ (when^{(1_1)} \ldots \circ when^{(p_1)})) \ldots \circ how^{(n)}(\ldots) \end{aligned} \tag{3}$$

A fragment *how* indicates an action towards realizing its associated objective, a fragment *who* presents who is carrying out the action, and fragments *when* and *where* describe the time and place of the action.

Note that Equation (1) expresses fragments that are associated with each other because they co-occurred in a response. Each clause *what* can have associated one clause *consequence* and multiple clauses *output* and alternative clauses *how* as indicated by symbol ... and distinguished by different superscripts. Symbol ... and the superscripts inside parentheses denote a sequence of multiple clauses of the same kind that are associated with the clause appearing before the parentheses.

Example: Let's consider the following statement: "Medicine should be affordable to everyone, so that people feel secure". Clause *what* is "Medicine should be affordable", clause *for who* is "everyone", and clause *output* is "people feel secure". In Equation (1), the objective is the composition of the three clauses, $what \circ output \circ for\ who$. The statement "Medicine should be affordable to young people and seniors, so that people feel secure" has two clauses *for who*, "young people" and "seniors", hence the objective is $what \circ (output \circ (for\ who^{(1)} \circ for\ who^{(2)}))$. Next, the statement "Medicine should be affordable to young people and seniors, so that people feel secure and live independently" has two clauses *output*, "people feel secure" and "live independently", hence the objective is $what \circ (output^{(1)} \circ (for\ who^{(1_1)} \circ for\ who^{(2_1)})) \circ (output^{(2)} \circ (for\ who^{(1_2)} \circ for\ who^{(2_2)}))$. Finally, the statement "Taxes by big business should pay for medicine" is an implementation for the above objective. Its clause *how* is "taxes pay" and its clause *who* is "big business". Hence, if this statement is a response to the previous statement during agent communication, the following Equation (1) can be set-up as follows: $what \circ (output^{(1)} \circ (for\ who^{(1_1)} \circ for\ who^{(2_1)})) \circ (output^{(2)} \circ (for\ who^{(1_2)} \circ for\ who^{(2_2)})) \leftrightarrow how \circ who$.

Finding an implementation for an objective, i.e., identifying the right part for the left side of Equation (1), justifies the meaningfulness of the solution, as the objective is achieved by the implementation. It serves as an argument for the correctness of the objective. Different reasoning styles can be used to find an implementation, which is a sequence of connected clauses between a *what* clause and an *output* clause. The consequence of the output results from inserting an agent's relevant exemplars [52,53] in the experience for the clauses of the response. A utility (meaningfulness) is associated with the outcomes. Different consequences and hence utilities might be reached by different agents, as their relevant exemplars (thus experiences) might be different. During their interactions, agents can identify implementations, outputs, and consequences accepted by all agents (convergence), or find reasons that justify for each agent its own chosen implementations, output, and consequences, which are different from those chosen by another agent (divergence).

The model describes each response using a symbolic representation, as illustrated in Figure 2. Response fragments (clauses) *what*, *for who*, *output*, *consequence*, *how*, *who*, *where*, and *when* include concepts (e.g., nouns and pronouns) linked through verbs. Each concept is described by a Structured Symbolic Representation (SSR), as shown in Figure 2a. The figure presents concept i with features (attributes) $f_i^{(k)}$. Some features correspond to physical attributes, such as an image, shape, or color. A concept or some of its features are activated using the following two rules: (i) Activating the concept subsequently activates its features $f_i^{(k)}$ too. (ii) Activating a feature always activates the entire concept. In addition, bi-directional links between the concept and its features, characterized by labels, describe the probability that activating one activates the other too: $def_{i,m}$ is the probability of activating feature $f_i^{(m)}$ when concept i is activated, and $up_{m,i}$ is the probability of activating concept i if feature $f_i^{(m)}$ is activated. Each feature is defined by a set of representative samples. Some concepts or features are connected to specific emotions.

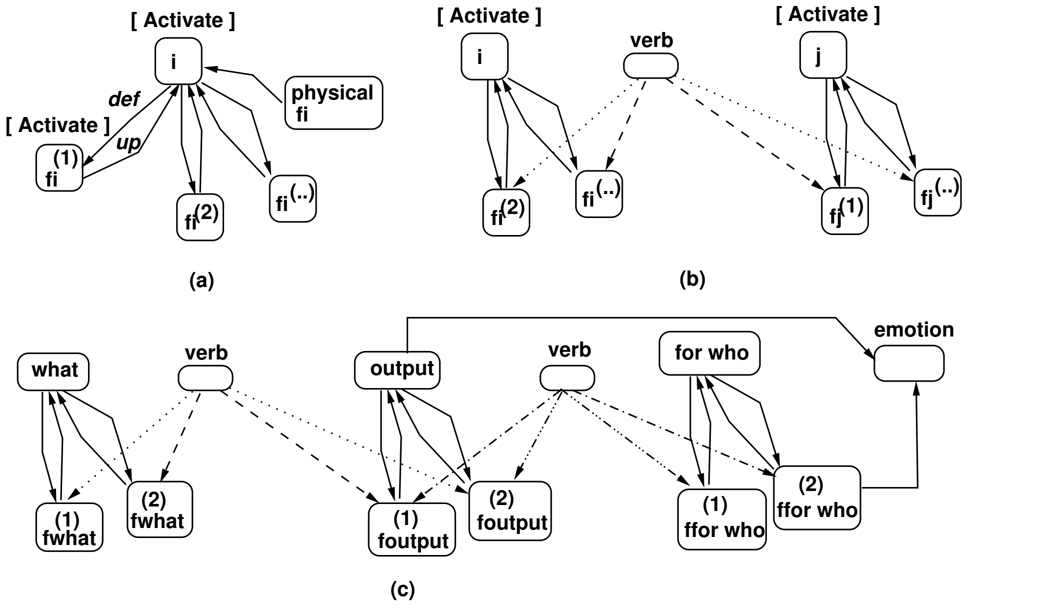

Figure 2. Response description: (**a**) Structured symbolic representation (SSR), (**b**) using verbs to connect concepts, and (**c**) description of the objective and implementation parts.

Verbs describe relations between concept attributes and concepts described as groups of features. The work of Fauconnier and Turner, among others, explains that verbs define different relations, i.e., part-whole, similarity, analogy, disanalogy, category, identity, representation, cause-effect, intentionality, role, property, time, and space [17]. The model groups relations into the following four semantic categories: property (P), identity (I), cause-effect (C), and role (R). Hence, a verb defines a mixture of relations pertaining to the four categories and involving different concept features:

$$verb =< \{rel_1^{Cat_P}, rel_2^{Cat_P}, ..., rel_x^{Cat_P}\}, \{rel_1^{Cat_I}, rel_2^{Cat_I}, ..., rel_y^{Cat_I}\}, \{rel_1^{Cat_C}, ..., rel_z^{Cat_C}\}, \{rel_1^{Cat_R}, ..., rel_w^{Cat_R}\} > \quad (4)$$

Label Cat_P refers to relations that describe properties, label Cat_I to relations for identity, label Cat_C to relations expressing cause-effect, and label Cat_R to relations presenting roles. Each value of rel_i is a verb's likelihood to represent a certain kind of relation among the properties of the concepts in a response. Some values rel_i result explicitly, if there is no doubt about their meaning in a clause or are found during response understanding if a response is ambiguous. Figure 2b illustrates how verbs connect concepts in an SSR. A verb connects features of the related concepts, such as feature $f_i^{(2)}$ of concept i and feature $f_j^{(..)}$ of concept j, and feature $f_i^{(..)}$ of concept i and feature $f_1^{(j)}$ of concept j.

3.2. Input Understanding during Agent Interaction

The proposed model for input understanding finds the best matching of a newly received response to the agent's experience and context while solving ambiguities about the input clauses, e.g., when the type of a clause is not fully decided. It finds the types of undecided clauses so that conflicts with the agent's experience and previous responses are minimized. Conflicts can relate to the logical consistency of the responses for more formal responses and to related emotional feelings in the case of ad-hoc responses. Input understanding is formulated as the following maximization problem:

$$\max \ \ clause\ matching \qquad \text{such that} \qquad \min \ \ resulting\ conflicts \quad (5)$$

Example: Let's assume that an LRGA receives the following response: "Medicare should cover nursing homes". Clause "nursing homes" could refer to a place (clause *where*), a time (clause *when*), or the recipient of an action (clause *for who*). Best matching assigns a clause type among the three possibilities (clauses *where*, *when*, or *for who*) that has the fewest conflicts with the agent's beliefs, experience, and previous responses. For example, if the discussion was about social categories such as students and the unemployed, then the clause type is more likely to be *for who* because another clause type would conflict with the context set-up with the previous responses. However, an agent's belief saying that "Government should support only elderly persons" suggests that the clause type should be *when*, as supporting any other social category also living in a nursing home would conflict with the belief.

Example: Best matching also considers ambiguities due to verbs. Let's assume a response states that "Seminars must teach physicians about new available things". The verb "teach" introduces relations between the attributes of the concepts "seminars" and "physicians". Even though a knowledge-related property of physicians, such as being well-trained or knowledgeable, is extended by the verb, it does not indicate if the verb indicates a new capacity (feature) of physicians (e.g., leaders, mentors, coordinators, etc.) or a specific procedure (i.e., performing a certain treatment). So, there is a certain degree of ambiguity introduced by the verb, and it must be solved by best matching. Similarly, clause *output* is not explicitly defined in this case, as the response only mentions that new things become available to physicians, even though these things will likely become available to their patients too.

Understanding input responses through the maximization in Equation (5) uses clause matching and stitching, as shown in Figure 3. Concept features are labeled as $f_{i,j}$ in the figure and pertain to the four categories, Cat_P, Cat_I, Cat_C, and Cat_R. Figure 3 also shows the relations rel_1, rel_k, and rel_y introduced by verbs $verb_1$ and $verb_2$ that connect the features of the three concepts (i.e., nouns) in the figure. Depending on the features involved in the most representative relations that connect it, a concept is characterized by its typical purpose, defined as the most frequent relations of its features. Some ideas about response understanding are also discussed in ref. [54].

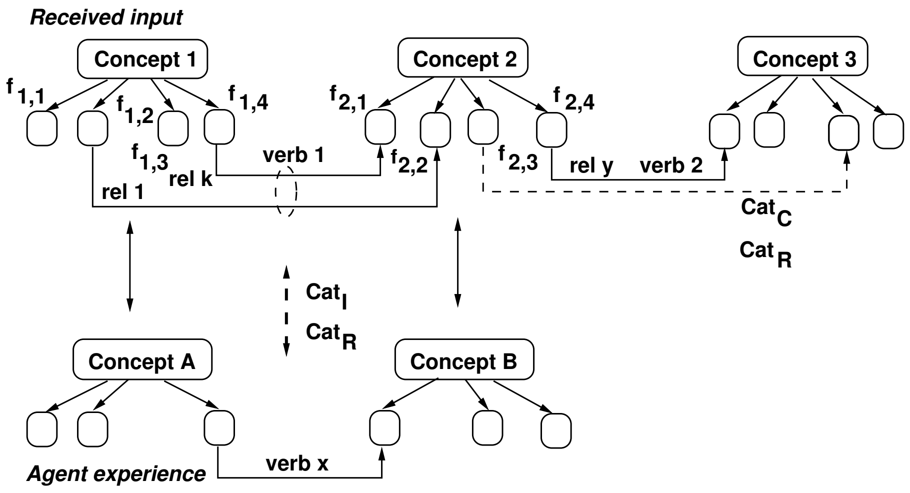

Figure 3. Matching and stitching of received input and agent experience.

Matching finds the correspondence between the concepts of an input and those in the agent's experience, e.g., concept *concept* 1 corresponds to concept *concept A*, and concept *concept* 2 to concept *concept B* in Figure 3. Concept $concept_i$ is matched to concept $concept_X$, if the following equation is maximized in an agent's memory:

$$matched(concept_i, concept_X) = \max_{X \in experience} \frac{similarity(concept_i, concept_X)}{\sum_{Y \in experience} concept_{i,Y}} \quad (6)$$

Y describes concepts in the agent's memory, and metric *similarity* describes the similarity of two concepts.

Stitching finds the maximum clause matching between the clauses of an input and an agent's own experience while minimizing the number of conflicts between the matched clauses. As a result, stitching assigns the most likely identities to any ambiguous relations in the input by finding unknown values $rel_i^{Cat_X}$ in description (4) (Cat_X is one of the four categories Cat_P, Cat_I, Cat_C, and Cat_R). Figure 3 illustrates the matching process. In this case, ambiguity relates to the fact that the types of the input clauses depend on the relations of verb $verb_1$, which can pertain to categories Cat_I (identity) or Cat_R (role). Similarly, verb $verb_2$ can pertain to multiple categories. A relation in category Cat_I is more likely to suggest a clause *what*, while a relation in category Cat_R a clause *how*. Verb $verb_x$ in the agent's own experience can be associated with verb $verb_1$ of the response or with verb $verb_2$. Stitching finds the most likely categories of unknown relations, and the sequencing order of the matched clauses (i.e., the maximum clause matching refers to verb $verb_1$ or to verb $verb_2$). Assuming a Bayesian model, the expected relationship category is as follows:

$$prob(rel^{Cat_X} | out, prec) = \frac{prob(out, prec | rel^{Cat_X}) \;\; prob(rel^{Cat_X})}{prob(out, prec)} \quad (7)$$

Hence, a response is understood by computing the probabilities in Equation (7), in which the frequencies of the entire experience of the LRGA or the frequencies of the input within a limited time window before the current response are used. Addressing ambiguous concepts and relations during matching and stitching is as follows. First, a prediction is made using a concept's typical purposes. Each concept is considered independently, and its observed features and relation types are used to predict the role of the concept. This prediction describes the most frequent cases. Similar to the real world [55], differences between expected results and observed results capture the attention of the agent. Furthermore, the agent starts to adjust matching and stitching to reflect the observed differences. The resolution of the ambiguities can, however, are changed by a later input in order to maintain the consistency of the entire sequence. For example, it is possible that the assumptions set for input ambiguities are changed by a subsequent input. Resolution uses matching and stitching between (i) two inputs with *what* clauses that are synonyms, similar, detailing, or generalizations of each other, (ii) two inputs with matched clauses *for who* and *who*, and (iii) two inputs in which matching uses *output* and *how* clauses.

3.3. Response Creation during Agent Interaction

Figure 4 presents agent interactions for a case in which agents agreed with each other as well as for two cases in which they disagreed. The figure shows how different kinds of clauses connect to each other during interaction while agents jointly solve a problem. It uses the following conventions. As it is not important in this example what each clause actually means (i.e., the words that form it), just their kind, the clauses were depicted as crossed boxes. However, to give more insight, some clauses were explicitly described by indicating their word structure. White boxes are clauses missing from Equation (1). Arrows between clauses, objectives, and implementations express associations, including operators $\circ$ and $\leftrightarrow$. The figure is based on the results of the cognitive experiment described in ref. [4]. Participants were randomly grouped into teams of four and asked to jointly devise solutions to improve the healthcare system in the USA. Each participant could provide an unlimited number of inputs, including responses to inputs offered by other team members.

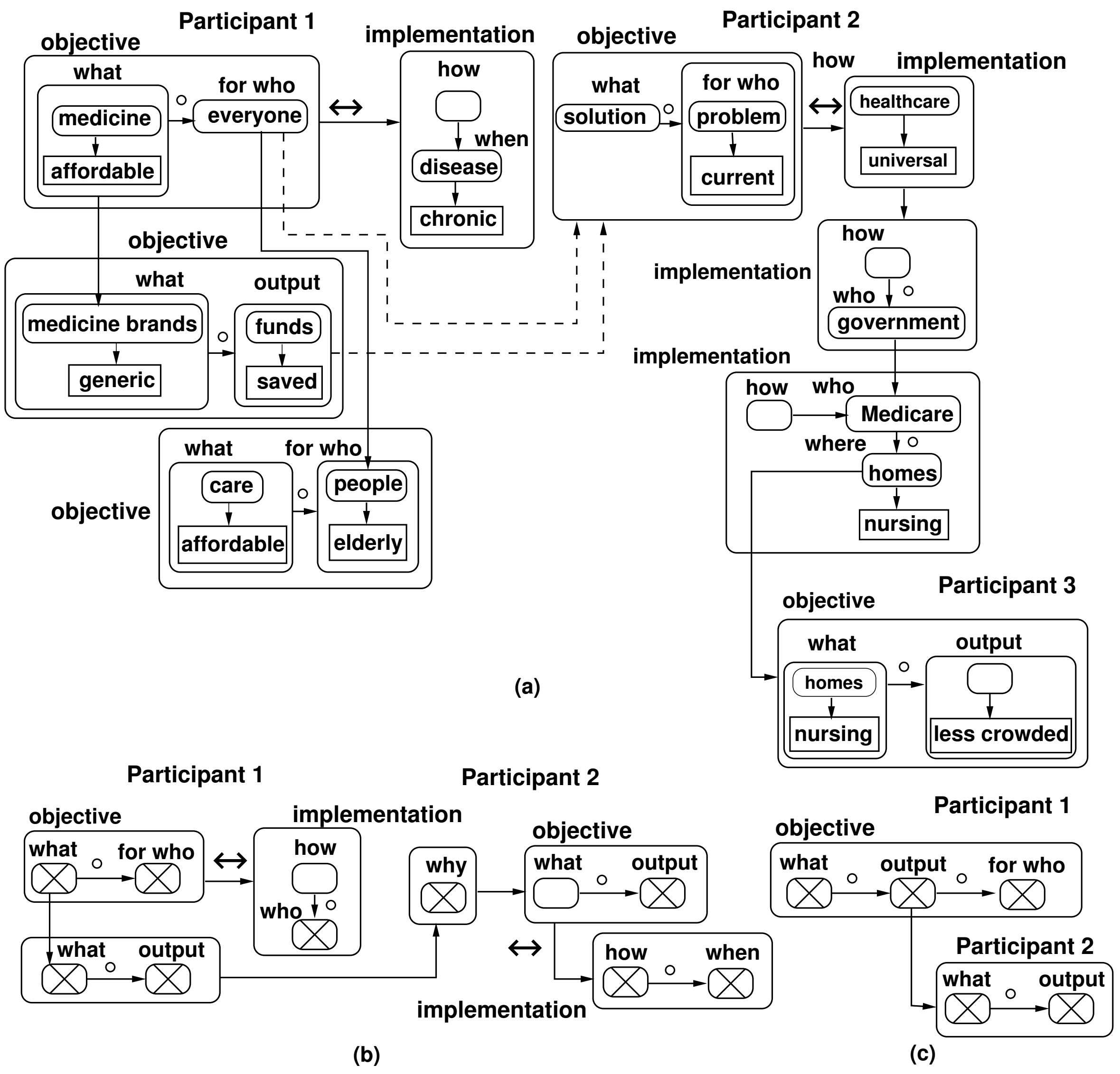

Figure 4. Generating a new response by (**a**) agreeing and (**b**,**c**) disagreeing participants.

In Figure 4a, members end-up agreeing on the proposed solutions to improve patient access to drugs. Three team members (Participant 1, Participant 2, and Participant 3) interact with each other. The interaction dynamics are as follows: First, Participant 1 suggests that "Medicine should be affordable to everyone with chronic disease". According to Equation (1), this input corresponds to the objective $medicine \circ everyone$ (in which clause *what* is "medicine" and clause *for who* is "everyone"), while the implementation includes only clause *when*, "chronic disease". Hence, clause *how* is missing, which is indicated by the white box in the figure. Participant 1 further details the objective by suggesting "generic medicine brands" (clause *what*) and saying that "funds are saved" as a clause *output* of the access to cheaper drugs. Another input by Participant 1 suggests that "affordable care should be offered" (clause *what*) "to elderly people" (clause *for who*). While Participant 2 does not directly continue the inputs of Participant 1, but it relates to the clauses of "elderly people" and "saved funds" by suggesting that "universal healthcare is a solution for the current problem". The figure shows these connections with dashed lines, as they are not directly related to drug access. The participant offers two implementation alternatives (e.g., right sides of Equation (1)), "government" (in which case only clause *who* occurs as clause *how* is unspecified) and "Medicare covering nursing homes". Participant 3 continues the clause "nursing homes" suggesting that "they should be less crowded". Figure 4a suggests that the interaction for agreeing members includes mainly detailing the referred clauses by adding more precise concepts, suggesting implementations for objectives, and presenting expected outputs. Participants are likely to use clauses or concepts said by others when creating new inputs.

Figures 4b,c show interactions in which the participants had opposing views. The interaction dynamics are different from the previous case. In Figure 4b, the disagreeing Participant 2 focuses on clause *purpose* (Equation (1)) and then elaborates on the related clause *output*. Participant 2 in Figure 4c uses clause *output* to suggest another solution (clause *what*). Hence, disagreeing participants focus less on elaborating objectives and implementations and more on clauses *purpose* and *output* that limit a solution. There are few responses that use clauses and concepts produced by others.

As explained at the beginning of Section 3 and shown in Figure 1, response generation can use a fast, top-down flow or a slow, bottom-up flow. The fast, top-down flow utilizes the input concepts that got high attention to inspect the knowledge representation in the agent's memory. The flow is controlled by the difference between the expected utility (the agent's previous response utility) and the actual utility (the input's utility), social interactions, and emotions [55,56]. The flow generates a response using the selected knowledge. The input prompts the agent's experience and goals. Furthermore, the agent compares the selected information in the memory with the input to find unforeseen concepts or clauses in the input. The most unforeseen concepts get the agent's attention [56]. These concepts are denoted as *dominant concepts*. Likely, they represent the cause of the unforeseen utility of the input (i.e., a higher or lower utility). The slow, bottom-up flow links and elaborates the selected knowledge (i.e., concepts, clauses, and verbs) to maximize the expected impact with respect to the intended goal. A new response combines experience and input items. It utilizes more abstract concepts, adds new symbols to describe a set of concepts grouped during previous, fast response generation, and uses simple insight obtained about concepts that caused a higher utility in preceding cases. Urgency selects among the top-down and bottom-up flows as discussed in Section 4.

Emotions influence the specific strategies (pattern) used in response creation. Figure 5 depicts the proposed modeling. Depending on the nature of agents' emotions, the classifier output shows four situations: confidence, positive surprise, negative surprise, and oppose. For a more detailed model, more situations can be considered at the classifier output to represent the similarity of the emotions. An agent's emotional response corresponds to certain perceived social situations (like reaching a consensus, disapproval, or approval or disapproval of the element of surprise) and triggers a specific kind of response pattern that creates the agent's output. Patterns are illustrated in Figure 5.

- *Confidence* corresponds to reaching a consensus among agents, indicating that they have similar perspectives (understandings) on a response, a response sequence, and their utilities (Figure 5a). As a result, agents create new responses that are motivations for the current *what-for who-output-consequence* clauses or create other outputs for the *what-for who* clauses for which exists agreement. As shown in Figure 5b, creating additional motivations reinforces the strength of the consensus reached about a response, its outcomes, and its consequences. Producing more outputs of similar utility adds more outputs and consequences to the current *what-for who* response clause. Figure 5c illustrates the situation. An existing output and consequence is further detailed by a subsequent response.
- *Positive surprise* indicates the situation in which an input includes a concept or clause that draws an agent's surprise reflecting its approval of the corresponding concept (Figure 5a). The utility expected by the agent is lower than the utility of the received input. The agent uses the element of surprise to create other examples (responses), or it can further detail the response including the element of surprise (Figure 5d). Also, the element of surprise can be generalized and then used in a new response. This is shown in Figure 5e in which the output of response (*what-for who* clauses) creates *what-for who* clauses of the abstraction, or *how* clause is generalized into a more abstract *what-for who* clause.
- *Negative surprise* describes the situation in which the concept or clause causing the agent's surprise is not valued by the agent (Figure 5a). The utility expected by the agent is higher than the utility of the received input. As a result, the element causing the surprise is utilized to motivate the separation of the new response from the input, such as in Figure 5f. Another subsequent response can generalize the element of disapproval (as in Figure 5e), and then use the generalization to motivate the disapproval through a *why* clause, such as in Figure 5f. The third option represents understanding the boundaries of the element of surprise. This is achieved by comparing the implementation (*how* frame) with the expected consequence of the output, as shown in Figure 5g.
- *Disapproval* corresponds to the situation in which agents have opposing understandings of the current response. Hence, the expected utility and the utility of the received input are opposite. As a result, the subsequent responses describe the following cases: motivation of the disapproval based on outputs and consequences (top of Figure 5h), explanation of the disapproval using motivation (*why* clause) (bottom of Figure 5h), and negation of *what-for who* clauses or their generalization followed by creation of an opposite alternative, e.g., new alternative *what-for who* clauses.

Example: Let's assume that the input response states the following: "More money should be spent to reduce false diagnosis". Furthermore, let's consider that after matching the input and the agent's experience during response understanding, the comparison points to "more money" as the dominant concept that separates the input from the agent's experience. The dominant concept gets the attention of the agent. Based on the matched experience, a certain emotion results as the dominant concept. This emotion is then integrated with the other emotion due to social interactions and the agent's present emotion. For example, an agent might consider that money must be spent carefully, so it does not accept the idea of increased costs. Hence, the emotion produced is anger. This emotion guides the generation of a fast, top-down response to the dominant concept. Anger selects the template to reject the response. The dominant concept gets combined with the template, followed by the prediction of the expected outcome and consequence. The agent's response indicates that higher expenses are unwanted.

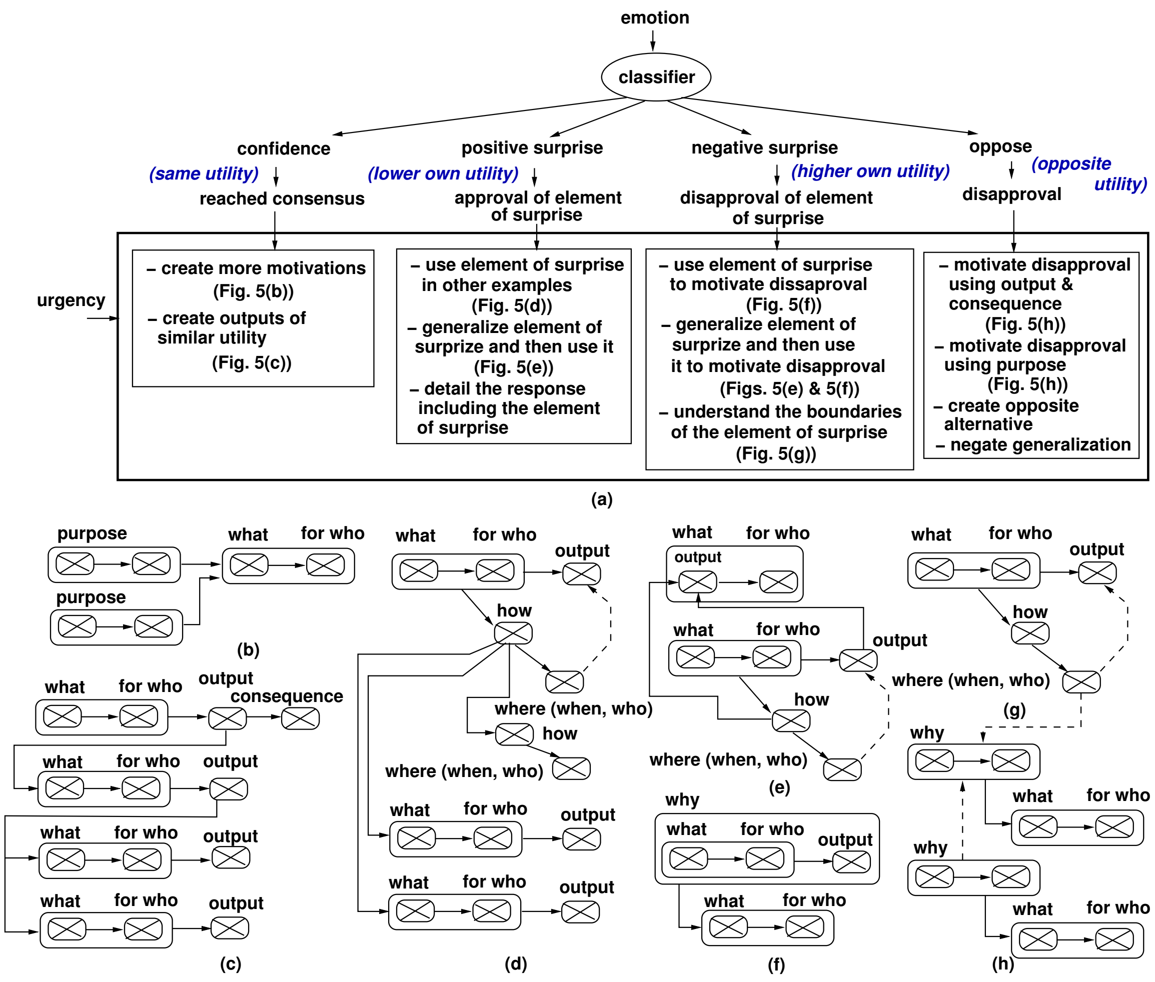
emotion
classifier
confidence
positive surprise
negative surprise
oppose
(same utility)
(lower own utility)
(higher own utility)
(opposite utility)
reached consensus
approval of element of surprise
disapproval of element of surprise
disapproval
urgency
– create more motivations (Fig. 5(b))
– create outputs of similar utility (Fig. 5(c))
– use element of surprise in other examples (Fig. 5(d))
– generalize element of surprize and then use it (Fig. 5(e))
– detail the response including the element of surprise
– use element of surprise to motivate dissaproval (Fig. 5(f))
– generalize element of surprize and then use it to motivate disapproval (Figs. 5(e) & 5(f))
– understand the boundaries of the element of surprise (Fig. 5(g))
– motivate disapproval using output & consequence (Fig. 5(h))
– motivate disapproval using purpose (Fig. 5(h))
– create opposite alternative
– negate generalization
(a)
purpose
what
for who
output
consequence
how
where (when, who)
why
(b)
(c)
(d)
(e)
(f)
(g)
(h)

Figure 5. Importance of emotions during response creation: (**a**) using emotions to select a response generation pattern, (**b**,**c**) adding motivations and outputs to reinforce a response, (**d**) detailing a response using an element of surprise, (**e**) generalizing an element of surprise, (**f**) generalizing an element of disapproval, (**g**) understanding the boundaries of an element of surprise, and (**h**) disapproval motivation.

4. Implementation of the Model

Figure 6 illustrates the software implementation of the LRGA model in Figure 1a. Every LRGA has a memory, implemented as a table, that stores the agent's experience (which is pre-loaded with a set of statements), the received inputs, and the responses created by the agent. Each memory entry represents a statement with the structure shown in Figure 1b. Each entry stores the related clauses, the nature of the clauses, the distinguishing word that draws the agent's attention, the attention strength associated with the clause, the emotions associated with the entire clause and with the distinguishing word, the utility of the clause, and if an agent disagrees with the statement in the entry (e.g., it has an opposite belief). Each memory entry (and its associated statement) is connected to other entries (and hence statements) in the agent's memory based on similar clauses *what*, different *what* but similar clauses *output*, similar *consequence* or implementations (synonyms), as well as entries expressing related statements at other levels of abstraction. A statement is considered more abstract than another statement if it is an objective for the latter.

First, an agent's implementation computes the difference between the received input and the agent's experience. The algorithm first searches the agent's memory to identify the most similar memory entry to the input. Clause similarity is the sum of similarities among the matched concepts of an entry and input, with an extra weight added to the similarity of the distinguishing words (the words that capture attention). The difference is in the distinct clauses between the input and the most similar memory entry. During finding similarities, the step also addresses the ambiguities of the input by implementing the maximization in requirement (5). It implements input understanding. More details about input understanding are offered in ref. [54]. The sources utilized to describe concept similarities and verb relations are as follows: The Merriam-Webster definitions of verbs were used to implement Equations (4) and (7). Also, WordNet descriptions [21] express the similarities between concepts. Relations for nouns and verbs were based on WordNet, such as descriptions of meronymy (part-whole relations), antonymy, attributes, similar meaning (polysemy), synonyms, entailment, and so on.

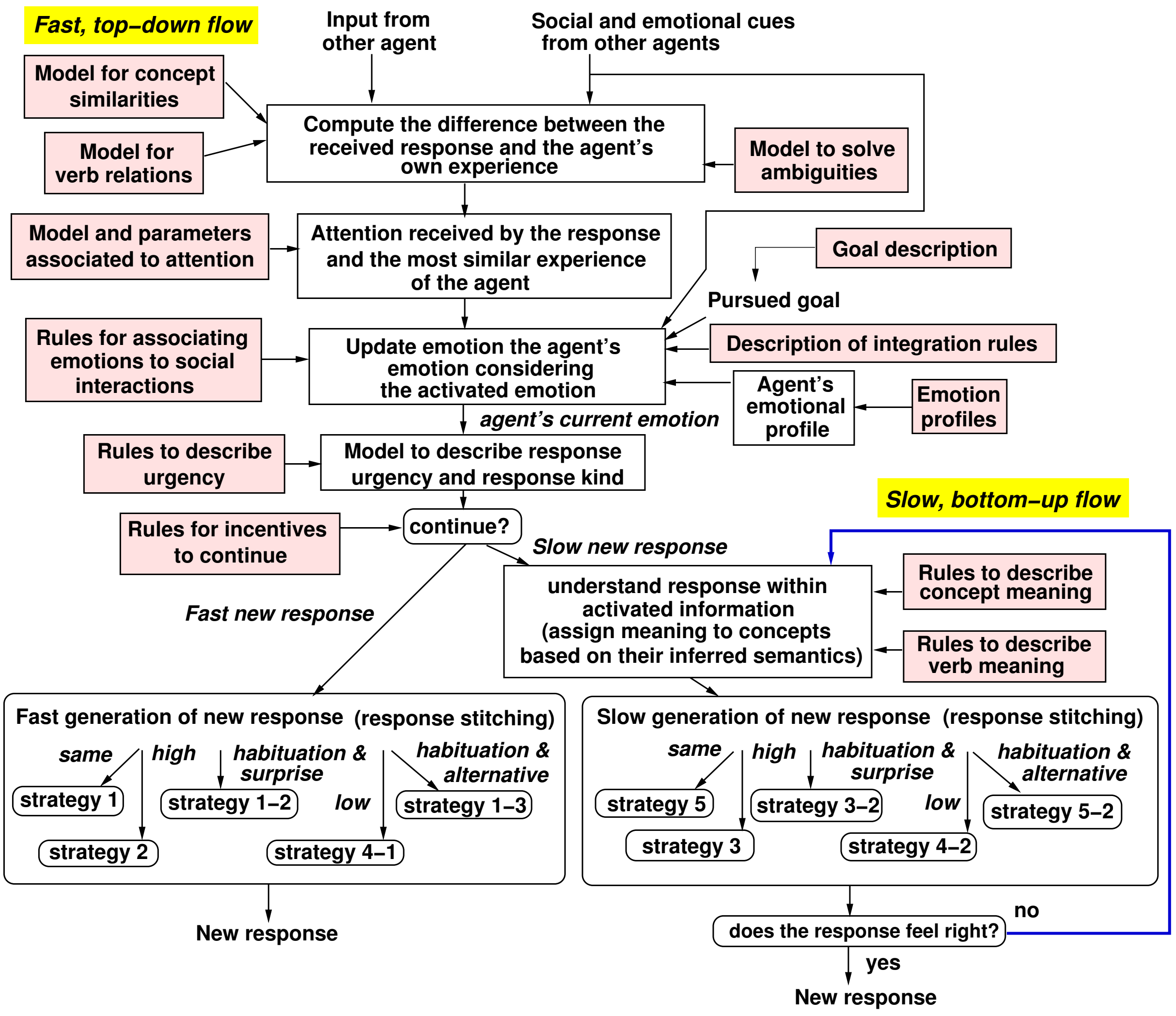
Fast, top–down flow
Input from other agent
Social and emotional cues from other agents
Model for concept similarities
Model for verb relations
Compute the difference between the received response and the agent's own experience
Model to solve ambiguities
Model and parameters associated to attention
Attention received by the response and the most similar experience of the agent
Goal description
Pursued goal
Rules for associating emotions to social interactions
Update emotion the agent's emotion considering the activated emotion
Description of integration rules
Agent's emotional profile
Emotion profiles
agent's current emotion
Rules to describe urgency
Model to describe response urgency and response kind
Slow, bottom–up flow
Rules for incentives to continue
continue?
Slow new response
Fast new response
understand response within activated information (assign meaning to concepts based on their inferred semantics)
Rules to describe concept meaning
Rules to describe verb meaning
Fast generation of new response (response stitching)
same
high
habituation & surprise
habituation & alternative
strategy 1
strategy 1–2
low
strategy 1–3
strategy 2
strategy 4–1
Slow generation of new response (response stitching)
same
high
habituation & surprise
habituation & alternative
strategy 5
strategy 3–2
low
strategy 5–2
strategy 3
strategy 4–2
New response
does the response feel right?
no
yes
New response

Figure 6. Software implementation of the LRGA model.

Second, the agent's attention is set on certain concepts of difference and the activated experience in reaction to the received input. The degree of dissimilarity of the involved concepts as compared to the previously used concepts is used to determine the strength of the drawn attention. The more unexpected a concept is, the higher the attention it draws. A set of predefined, parameterized rules establishes the connection between the dissimilarity value and the attention strength.

Third, an agent's emotion update is found by integrating the emotions associated with the activated experience, the emotion due to the social interaction cues, and the agent's current emotion. The emotions associated with the received inputs, i.e., after understanding the inputs, are similar to those represented by appraisal theory [48,49]. Appraisal models present emotions as a result of an agent's expectations, which are the relationship between its own goals, beliefs, and desires and the observed events, i.e., the received responses.

Specifically, emotion integration performs three steps: The first step combines the emotions of the most similar clause in the memory and its distinguishing word using a set of predefined rules. Furthermore, the resulting emotional cue is integrated with the emotional aspects associated to the social interactions with the other agents, depending on the nature of social coordination and compatibility between agents [57]. The used rules reflect situations in which agents have a higher degree of compatibility between their experiences and goals as well as agents with a high degree of divergence their experiences and goals. The implementation considers the following interaction types and their associated emotions: neutral, desire to lead, desire to not fall behind in the amount of provided responses, desire to imitate (to fit with the rest), and desire to contradict. All but type neutral have two versions, a moderate and a strong variant. The last step integrates the emotional cue fromthe previous step with the agent's current emotion to produce any changes in emotion (i.e., an update in emotion). A predefined set of rules is used for this step too.

An agent's emotion profile in Figure 6 is realized as follows. Each agent can have fifteen different emotions, including neutral, happy, angry, fear, trust, and so on. Agent emotion profiles are set-up when agents are created and do not change during execution. Agent emotion profiles capture two aspects:

- The change of emotion strength (valence) over time: Figure 7a shows the variation of the strength of an emotional state over time. The parameters of the strength plot (e.g., values $s0$, $s1$, $s2$, $s3$, $t1$, $t2$, $t3$ defining the slopes of the linearized plots) are specific to each agent and set-up when the agent is created.
- Emotion transitions during interactions: The predefined rules for emotion integration change emotions as shown in Figure 7b. Each emotional state has sub-states, e.g., *state-1*, *state-2*, *state-3*, which correspond to the emotion strength in Figure 7a. Transitions between sub-states are controlled by cues (c) and the passing of time (t). The nature of the transitions and their controlling functions $F(c,t)$ (describing conditions) are specific to each agent.

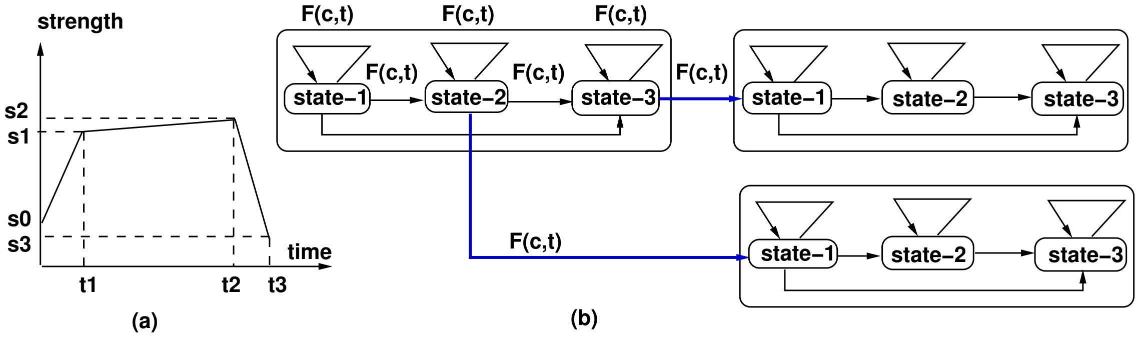

Figure 7. Emotion implementation: (**a**) emotional state strength variation over time and (**b**) rule definition for emotion integration. Parameter c is a cue, parameter t is time, and $F(c,t)$ is the controlling function.

The fourth step determines the urgency, the way in which the agent treats received input, and the kind of response it will generate as a result. Depending on the similarity and familiarity (i.e., number) with previous responses, an agent creates a new response through a fast, top-down path or a slow, bottom-up path. Initially, the top-down path is pursued, but after a number of related responses, an agent switches to the bottom-up path to produce a response that generalizes the previous. Furthermore, the difference between the utility of the received input (which describes the utility assigned by the other agents) and that of its own response (which describes the expected utility) decides the way (i.e., the strategy) in which the agent creates its new response. The current implementation distinguishes between five different situations: (i) same utility, (ii) higher actual utility than expected, and (iii) lower actual utility than expected. In addition, for the same utility case, there are two more situations that describe the habituation of the agent with the current type of responses (e.g., the sequence includes several similar responses): (iv) habituation and surprise indicate that the response contains an unexpected concept but the utility remains the same, and (v) habituation and alternative suggest that the agent explores clause alternatives that keep the expected utility the same.

The fast, top-down response generation flow uses an agent's most similar experience to the received input to produce a new response. Fast response understanding identifies the main concepts involved and assigns their meaning (semantics) based on their usual usage without solving input ambiguities. Hence, clause meanings depend only on the frequencies of the different types the clause had previously, and there is no connection to the broader context in which the response was produced. A fast, new response is created by stitching together the activated abstract pattern and the main concepts of the response, which are those that drew the agent's attention. Depending on the degree of urgency, the fast response is the output, or response generation continues with the slow, bottom-up process shown in the right part of Figure 6.

The fast strategies used to assemble a new response are selected depending on the difference between the actual and expected utilities of the responses, as shown in Figure 6. Strategy 1 builds a new response that extends the agent's most similar experience to the received response by adding details to the response, e.g., clauses that are missing. It incrementally produces responses for the situation, if the actual and expected utilities are the same, but the agent does not feel comfortable moving beyond its own experience (hence, producing only a small number of similar responses). Strategy 1-2 and Strategy 1-3 correspond to situations of habituation, when the agent is sufficiently comfortable to explore beyond its immediate experience, such as by extending a previous experience by considering alternatives to the concepts or by incorporating unexpected concepts present in the received responses. Strategy 2 is executed when the actual utility is higher than expected. As a result, the agent creates a response that attempts to bridge its own experience to the received response of higher utility. To achieve it, it combines the different clauses of the response (which are considered to be the reason for the higher utility) with the clauses of its own experience. Strategy 4-1 describes the case in which the actual utility of the response is lower than that of the expected utility, e.g., due to its own experience. As a result, the new response of the agent further emphasizes the differences between its experience and response, as the differences are considered to be the causes of the difference.

The slow, bottom-up response generation flow proceeds as follows. The received response is further understood by assigning meaning to the ambiguous concepts, as detailed in ref. [54]. Furthermore, a new response is created based on the pursued goals and by stitching clauses of the received input with clauses of the agent's experience, depending on the kind of difference between the actual utility of the response and the expected utility as represented by the agent's experience. If the two utilities are similar and no habituation has occurred yet, the agent attempts to generalize its own experience using Strategy 5. It is assumed that the previous fast steps produced enough details to support generalizations. Similarly, if habituation has already occurred, the agent generalizes the concept for which alternatives were produced during the preceding fast response steps. If the actual utility is higher than the expected value, the agent uses Strategy 3 to bridge, at the abstract level, the differences between a response and an experience. Abstract concepts are used by the agent from the response and experience to create a new response. Strategy 3-2 is a variant of Strategy 3, in which an abstraction of the unusual response clause is combined with the abstract clauses of the experience. The strategy is used when an unexpected clause is present in the response after habituation has occurred. Strategy 4-2 presents the case in which the actual utility is less than expected. The agent response focuses on generalizing the differences between the response and experience and on expanding a clause *how* for the difference in order to support the validity of the different clauses being the reason for the difference in utilities. The problem-solving process can continue to generate more responses until it is considered satisfactory (as shown by the blue arrow in Figure 6).

5. Experimental Results

Experiments studied the characteristics of agent interactions, e.g., the responses created by two agents that exchange a sequence of responses. The main objective of the experiments was to analyze the variety of responses, such as clauses that were propagated during agent interactions and those that were filtered out, experience items that were reinforced, knowledge detailing and abstraction, the nature of created alternatives, i.e., synonyms and homonyms, the creation of bridging responses between knowledge items that were initially unrelated, and the separation of the knowledge items of two agents. Experiments also measured the frequency of using the different strategies (ways) of generating responses.

Experiments used a knowledge base that was set up using seventy-five responses produced by the group experiments with human subjects discussed in ref. [4]. The study examined the relationship between group diversity and group creativity using a problem that required groups to improve the healthcare system in the USA. Each group included five human participants who collaborated by exchanging ideas in an asynchronous fashion. Seventy-five randomly selected responses from this study were used to set up the agent's memory in our work. Each response was a sentence with nouns, verbs, pronouns, adjectives, and so on. Each agent's memory included twenty randomly selected responses out of the seventy-five responses. Some of the twenty responses were shared by multiple agents to create shared experiences among agents. The similarity of concepts was characterized using metrics from WordNet [21]. All experiments used two interacting learning agents. Experiments were conducted using an Inspiron 15 5510 laptop with an Intel i7-113770H processor at 3.3 GHz and 16 GB of RAM.

5.1. Characteristics of Agent Interactions

The execution time depends on the number of responses produced by the interacting agents. It takes around 1.1 s to generate one response; therefore, execution time stays low for the considered team sizes (up to four agents) and number of responses (up to a few hundred responses). Finding the most similar clause in an agent's memory to a received input is computationally the most intensive routine. Its complexity is linear with an agent's experience memory size and quadratic with the number of clauses in a response. Nevertheless, complexity remains low, as a response can have up to nine clauses in this model. Hence, the implementation scales well with the number of agents and responses.

Table 1 summarizes the ratio of the different response types (e.g., same utilities, same utilities with habituation, same utilities with alternatives, lower utilities, and higher utilities) as well as their two versions, the fast version and the slow version. The table presents the number of times the different cases occurred out of a total of 693 runs and the corresponding percentage in parentheses. Note that these values depend on the software parameters used in the experiments, such as the total number of interactions between the two agents, the utility threshold for which two utility values are perceived to be similar, the number of iterations before switching from the fast version to the slow version used in creating responses, and the number of iterations before switching towards procedures that encourage more response diversification, such as same utilities with habituation and same utilities with alternatives.

Table 1. The ratios of the utilized response construction procedures.

Agent	Same Utilities	Same Util. with Habit.	Same Util. with Alter.	Lower Utilities	Higher Utilities	Fast Version	Slow Version
Agent 1	68 (9.81%)	47 (6.68%)	19 (2.74%)	124 (17.89%)	108 (15.58%)	205 (29.58%)	160 (23.08%)
Agent 2	54 (7.79%)	39 (5.62%)	20 (2.88%)	109 (15.72%)	105 (15.15%)	170 (24.53%)	158 (22.79%)
Total	17.6%	12.3%	5.62%	33.61%	30.73%	54.11%	45.87%

Even though it was not our intention, the experimental results in the table show that there were similar ratios encountered for cases with the same utilities (including routines for same with habituation and same with alternatives), lower utilities, and higher utilities of the received inputs, and most similar experiences of the agents. Fast versions of response creation were slightly more frequent than slow versions. Routines for same with habituation and same habituation with alternatives were used much less than the same utility response generation procedure.

We next analyzed the nature of the responses that were created by using the response construction procedures summarized in Table 1. Table 2 summarizes the results of the analysis. The rows correspond to the five kinds of procedures (the rows of Table 1). Columns present the following information: the first two columns describe the reduction and extension of clause ambiguity, e.g., the same concept can serve multiple clauses in an input. Column (a) indicates the learning of an input, i.e., a received input is copied unchanged into an agent's experience memory. Column (b) presents the reinforcement of an existing experience item, such as when the produced response exactly matches an existing experience item. Column (c) describes the cases in which the created response is an instance of the input, and Column (d) describes the situations in which the response is an instance of an existing experience item. Column (e) represents responses that are synonyms of the received inputs, e.g., the two share the same *what* clause and some of the other clauses but also have different non-*what* clauses. Column (f) indicates the number of responses that were synonyms of an existing experience item. Column (g) presents the number of responses that were homonyms of inputs, i.e., the two shared non-*what* clauses but had different *what* clauses. Column (h) displays the number of homonyms for existing experience items. Column (i) shows the percentage of responses for which inputs were instances, and Column (j) shows the percentage of responses for which an experience item was an instance. The last column describes the number of responses that were unrelated to the input (i.e., did not have a common clause or concept) and hence separated the response from the input. Finally, columns Abs-1–Abs-5 summarize the nature of the created abstractions through the responses generated by slow versions of the procedures. Column Abs-1 presents the percentage of cases in which a new symbol was introduced as a result of slow flow. Column Abs-2 indicates the percentage of cases in which the response was an abstraction of a previous abstraction. Column Abs-3 shows the percentage of cases, in which the input was an instance of the response, and Column Abs-4 shows the percentage in which an experience item was an instance of the response. Column Abs-5 indicates the percentage of cases in which the response was an abstraction of a clause of an experience item.

Table 2. The ratios of the different types of responses.

	Red. amb. (%)	**Ext. amb. (%)**	**(a) (%)**	**(b) (%)**	**(c) (%)**	**(d) (%)**	**(e) (%)**	**(f) (%)**	**(g) (%)**	**(h) (%)**	**(i) (%)**	**(j) (%)**	**Abs-1 (%)**	**Abs-2 (%)**	**Abs-3 (%)**	**Abs-4 (%)**	**Abs-5 (%)**	**Sep (%)**
same	4.8	0.0	11.6	28.5	12.3	14.0	5.9	0.5	4.3	1.0	8.0	14.0	N/A	N/A	N/A	N/A	N/A	N/A
same habit.	2.0	1.0	2.0	37.0	2.0	0	3.0	0.0	3.0	0.0	19.5	31.0	N/A	N/A	N/A	N/A	N/A	N/A
same alter.	0.0	7.1	0.0	35.0	0.0	21.4	14.2	7.1	7.1	28.5	0.0	214	N/A	N/A	N/A	N/A	N/A	N/A
lower	7.2	2.5	3.6	7.2	11.3	5.1	3.0	3.6	3.6	7.73	12.3	13.9	5.29	1.96	1.96	5.8	7.8	N/A
higher	15.1	0.45	1.83	18.3	0.45	3.2	9.63	4.58	7.33	1.83	7.79	19.26	N/A	N/A	2.43	2.43	N/A	8.25

Table 2 shows that most ambiguity reductions resulted in response generation for higher and lower utilities. While ambiguity reduction does not depend on the actual generation method, it is called for in the slow response generation flow. This suggests that there was a higher percentage of slow flow for higher and lower utilities than for the other three response generation situations. Furthermore, there was a slight increase in ambiguity, mainly for generation for the same utilities with alternatives. Combining clauses present in the input but not in an experience item can increase ambiguity if the concept of the input clause is already present in the experience item. Learning an input is relatively rare, with the exception of methods for the same utilities (11.6%). Reinforcing an existing experience entry is likely to occur in all situations (18.3–37%), with the exception of the lower utility case, as it attempts to create a response that is more similar to a higher utility input. Generation for same and lower utility is more likely to create responses that are instances of an input (12.3% and 11.3%, respectively), and generation for same (14.0%) and same with alternatives produces instances of inputs (21.4%). More responses that are synonyms of inputs are generated for the same utility with alternatives (14.2%) and higher utility (9.63%). There are rare responses that are synonyms of experience items, with a maximum of 7.1% for the same utility with alternatives and 4.58% for higher utility. Similarly, most homonyms of inputs are for the same utility with alternatives (7.1%) and higher (7.33%). Most homonyms for experience items can occur for the same utility with alternatives (28.5%) or lower utility (7.73%). Response generation for lower utility created a new symbol in 5.29% of the cases, in 5.8% an abstraction of the experience item, and in 7.8% an abstraction of a clause in an experience item.

The next two experiments studied the degree to which the different kinds of response generation methods changed the knowledge space of the two agents involved in the interaction. The metrics captured the change in the strength of connectivity between related concepts and the degree to which unrelated concepts get connected through a sequence of similar concepts.

Metrics were computed based on the length of the shortest path linking two concepts in a graph, in which each concept is a node and an arc connects two concepts with similarity beyond a minimum threshold. Similarity labels every arc and were set using WordNet similarity values [21]. Dijkstra's algorithm was used to find the shortest paths.

Table 3 presents the results for fast response generation flows, and Table 4 indicates the results for slow versions. Rows Avg., max, and min Δ total length show the average, maximum, and minimum change of the total length of the paths that link any two connected concepts after performing the corresponding response generation method. Rows Avg., max, and min Δ average length per node offer the average, maximum, and minimum values of the ratio of the total path lengths and the number of pairs of connected nodes. Rows Avg., max, and min Δ adjusted total length indicate the change due to a response generation step to the average, maximum, and minimum adjusted total length, which is the total length of all node pairs in a graph. A large penalty value was introduced for two unconnected nodes. Rows Avg., max, and min Δ average adjusted length per node present the change in the ratio of the average, maximum, and minimum adjusted total length and the total number of node pairs in the graph. Note that for the metrics using adjusted total length (last six rows), an improvement is indicated by negative values, as decreases in the adjusted total lengths show that previously unconnected nodes (which add a large penalty to the path lengths) were linked by the new responses.

Table 3 indicates that for response generation, same utility and same utility with alternatives offer the highest improvement in the connectivity of already connected nodes, as shown by average and maximum values. However, the same utility with alternatives produced a small overall impact, as it occurred only rarely. The smallest improvement was for lower utility as it attempted to connect concepts that were not so well linked together. A higher utility situation created a slightly better improvement, likely because it was not attempting to tie together unconnected concepts. Instead, it separated them. Each response generation case produced similar minimum improvements, and each case showed a large range of changes between the minimum and maximum values.

Experimental results suggest that generation for the same utility mainly introduces local connections between non-*what* clauses. Responses were also likely to have more clauses, which favored the likelihood of improving connectivity. Generation for lower utility created more global connections as they combined different *what* clauses, therefore different concept sub-graphs that correspond to these. Experiments also showed that generation using the same utility created a certain difference between the results for the two agents: there was a higher improvement for Agent 1 and a smaller improvement for Agent 2. In contrast, lower and higher utility situations produced similar results for the two agents. These results suggest that lower and higher utility cases were more systematic in producing improvements for both agents, while the same utility flows can hit on responses that offer high improvement, but this is not guaranteed for the other agent too. Generation cases that have higher experience with item reinforcement are likely to have less improvement, as reinforcement does not change concept connectivity. A low-utility situation produces responses that always connect input clauses to non-*what* clauses of the experience but can offer fewer improvements if the input clause has a different *what* clause. However, these connections can lead to a more effective linking of previously unconnected concepts, as shown by metrics related to adjusted total length. Responses produced by higher utility flows show a small separation from other concepts, which could lead to the partitioning of an LARG's knowledge space.

Table 3. Performance metrics for fast versions.

Metric	**Same Utility**		**Same Utility with Habituation**		**Same Utility with Alternatives**		**Lower Utility**		**Higher Utility**	
	Agent1	**Agent2**	**Agent1**	**Agent2**	**Agent1**	**Agent2**	**Agent1**	**Agent2**	**Agent1**	**Agent2**
Avg. Δ length per node	109.38	59.56	53.34	91.12	132	52.67	65.29	66.71	71.82	69.44
max Δtot. length	287.94	110.29	74.0	200.0	264.8	81.0	193.05	136.17	118.0	130.15
min Δtot. length	33.0	14.89	30.0	21.75	55.3	26.41	26.4	8.0	30.0	2.25
Avg. Δ avg. length per node	0.041	0.011	0.016	0.045	0.053	0.016	0.025	0.025	0.014	0.022
max Δ avg. length per node	0.184	0.021	0.063	0.012	0.133	0.006	0.108	0.057	0.023	0.088
min Δ avg. length per node	0.01	0.01	0.01	0.223	0.016	0.008	0.0001	−0.003	0.01	0.001
Avg. Δ adj. total length	−0.17	0.046	0.0076	0.016	−0.24	0.025	−0.021	−0.008	−0.02	−0.031
max Δ adj. total length	−0.81	−0.09	−0.002	−0.13	−0.61	0.07	−0.35	−0.39	−0.004	−0.49
min Δ adj. total length	−0.24	0.13	0.0001	0.0001	−0.01	0.014	0.14	0.0001	0.14	0.17
Avg. Δ avg. adj. length per node	−0.014	0.0007	0.0236	0.016	−0.016	−0.009	−0.013	−0.024	−0.013	−0.015
max Δ avg. adj. length per node	−0.04	−0.005	−0.005	−0.093	−0.017	−0.013	−0.028	−0.002	−0.009	−0.013
min Δ avg. adj. length per node	−0.01	0.0019	0.013	0.0026	−0.014	−0.005	0.003	−0.0001	−0.0006	−0.002

Table 4 presents the same metric values for slow-response generation flows. At a more abstract level, the lowest utility flows produce the best improvement for the already connected nodes, while the same utility flows create the newest links for unconnected nodes. The first observation is expected, as combining at higher levels any already connected concepts can increase the total connectivity. The latter observation can be explained by the fact that combining abstract concepts such as slow, same utility flows has a similar, global effect on the knowledge structure as fast, lower-utility flows. Slower utility flows can introduce new symbol that are less connected to existing concepts.

Table 4. Performance metrics for slow versions.

Metric	Same Utility		Same Utility with Habituation		Same Utility with Alternatives		Lower Utility		Higher Utility	
	Agent1	**Agent2**	**Agent1**	**Agent2**	**Agent1**	**Agent2**	**Agent1**	**Agent2**	**Agent1**	**Agent2**
Avg. Δ total length	131.25	129.46	145.35	90.41	96.18	118.45	234.17	154.82	191.58	125.58
max Δ total length	267.0	204.0	308.0	257.0	161.0	187.0	352.0	294.0	265.0	320.15
min Δ total length	14.0	24.0	5.0	25.0	29.0	37.0	18.0	48.0	34.0	14.0
Avg. Δ avg. length per node	0.0049	0.0076	0.0033	0.003	0.0087	0.02	0.0056	0.006	0.0091	0.008
max Δ avg. length per node	0.025	0.04	0.015	0.017	0.03	0.059	0.055	0.012	0.056	0.09
min Δ avg. length per node	−0.002	0.001	−0.007	−0.002	−0.004	0.009	−0.005	−0.001	−0.003	0.001
Avg. Δ adj. total length	0.0043	0.0364	0.0141	0.042	0.021	0.16	−0.0011	0.01	−0.0029	−0.0082
max Δ adj. total length	−0.13	−0.26	−0.14	−0.59	−0.11	−0.24	−0.14	−0.13	−0.23	−0.15
min Δ adj. total length	0.12	0.17	0.008	0.49	0.25	0.07	0.12	0.009	0.11	0.11
Avg. Δ avg. adj. length per node	−0.0034	−0.0011	−0.002	−0.004	−0.0075	−0.0071	0.0022	−0.0097	−0.0014	0.0015
max Δ avg. adj. length per node	−0.004	−0.008	−0.004	−0.0055	−0.019	−0.015	−0.087	−0.0018	−0.006	−0.0053
min Δ avg. adj. length per node	−0.0006	−0.003	0.0009	0.0007	−0.001	−0.003	−0.0005	−0.0003	−0.0007	−0.0001

5.2. Qualitative Comparison to Agents Using Commitment-Based Protocols

This subsection offers a qualitative comparison of the proposed model, LRGA, and models using commitment-based protocols. These models were discussed in Section 2. As their foundations are quite different, a numerical comparison is difficult. The qualitative comparison uses the Gold Miners Scenario Problem (GMSP) [30], a traditional benchmark case for studying commitment-based protocols. The problem is that gold nuggets are placed on a two-dimensional grid with obstacles. A team of interacting agents collects gold nuggets as they move along the grid and avoid obstacles. An agent can pick up a gold nugget to carry it to a depot, drop a nugget, communicate about a nugget to another agent, or just ignore a nugget and move on. The goal is for a team to maximize the amount of gold brought to the depot. Table 5 summarizes the comparison.

Like reaching consensus or disagreement, GMSP describes a team-based problem-solving situation. The qualitative similarities and differences between the two are as follows. The goal of GMSP is to collect as much gold as possible, which is conceptually similar to finding many common ideas in the LRGA. Similarly, the distance over which a gold nugget must be carried to the depot is similar to the dissimilarity of two ideas that must be connected to each other by the agents' responses. The GMSP actions and the response generation situations are similar in that they include incremental changes to the current situation, such as moving to a neighboring grid cell in GMSP and creating an incremental response by adding or changing a clause (Strategy 1). However, while GMSP uses a static neighborhood definition in which every grid has four well-defined neighbors, the clauses that can be changed in a response are dynamically decided by each agent based on its experience and urgency. Moreover, clauses can be ambiguous. In addition, LRGAs can create responses using strategies that go beyond incremental changes, such as incorporating unexpected clauses into a response. This would be similar to a mining agent in GMSP suddenly jumping into a region that lies between two grid regions that were already explored by the agents. While in GMSP it would be disadvantageous if an agent explored the same grid region as another agent, repetition in LRGA might indicate reinforcement of a statement and hence increase its priority. Carrying a gold nugget is similar to producing a response that relates to the current ideas, while dropping a gold nugget is arguably similar to dropping the current idea pursued by an agent to follow an unrelated idea. Communicating to others about a gold nugget is similar to communicating a response to all LRGAs. A difference is that the process towards reaching a consensus or disagreement involves responses from all agents, which is similar to agents jointly carrying a gold nugget to the depot. However, in GMSP, one agent can carry one nugget at a time. Moreover, all agents in GMSP perceive the grid and gold positions in the same way, while in LRGA, the actual meaning of a response is influenced to some degree by the agent's experience, e.g., the experience is used to solve any clause ambiguities. Therefore, in GMSP, increasing the amount of communication increases the amount of shared knowledge, i.e., the communicated positions of gold, while in LRGA, increasing the amount of communication can lead to disagreement. Finally, interactions through commitments impose a certain set of actions depending on an agent's role, while LRGAs do not have roles; instead, their responses are generated using a set of strategies that are dynamically selected based on the agent's experience and urgency. Therefore, in addition to being grounded in theories in psychology and sociology, the LRGA model considers a different semantics of the problem and interaction space, which is the semantics of statements, as well as another approach to producing the responses through which agents interact with each other.

Table 5. Summary of the qualitative comparison between LRGA and commitment-based protocol agent model.

	Commitment-Based Protocol Agents	**LRGA**
Studied problem	Gold nuggets distributed on a 2D grid with obstacles	Agents create and communicate responses on improving a situation
Goal	Bring as many gold nuggets as possible to the depot	Reach consensus or disagreement about an idea
Theoretical grounding	Formal logic	Psychology and sociology for agent behavior, linguistics for responses
Problem parameters	Distance of nuggets from the depot	Dissimilarity of ideas and agent experiences
Solution space	Static neighborhoods based on four neighbors	Dynamically generated responses
Agent actions	Navigate grid, avoid obstacles, pick-up nuggets carry nugget, drop nugget, communicate about positions	Understand response, add details to a response, consider alternatives with same meaning, abstract, include rare concepts, combine abstractions, communicate created responses
Communication mechanism	Static, based on agent roles	Dynamic based on agent expectations and urgency
Solution continuity	Keep carrying a picked nugget, move to neighbor	Continue current idea by incremental changes
Solution discontinuity	Drop nugget [no jumps allowed]	Adopt new unrelated idea to create own response
Situation interpretation	All agents perceive agent and nuggets positions in the same way	Ambiguities and different utilities can exist for the same inputs
Agent collaboration	Communicate positions, cannot jointly carry a nugget	Communicate ideas, agents can change the same idea
Repetition	Always sub-optimal	Can be beneficial

6. Discussion of LRGA Model

This section discusses the LRGA model as compared to similar MAS models presented in Section 2. MAS models have been traditionally used for distributed problem solving, like cooperative search in GMSP to maximize a total reward [44], numerical optimization, e.g., bean design [35], or to study feature spreading in populations [36]. In contrast, the LRGA model was devised for problems that require generating new ideas that are accepted by participants with partially-different assessment values due to different goals, utilities, and experiences. Problem framing, common ground identification, and HCI for self-improvement and teaching are such problems. The solution space of these problems must be dynamically constructed through agent interaction without using a closed-form description, like in GMSP [30,31] and in ref. [36], or cost functions, like in engineering trade-off exploration [34,35]. Moreover, in the LRGA model, as presented in Section 3, the exchanged ideas during problem solving are networks of symbolic fragments structured based on language characteristics and not numerical [36], mathematical, or logic expressions as in most related MAS [14,32,41]. Ideas can be ambiguous due to the inherent nature of language; hence, agents might differently interpret inputs depending on their specific experience, attention, emotions, and assigned utilities (priorities). We think that the existing MAS models cannot tackle these problem specifics well.

Similar to KABOOM [34], the first MAS to consider the cognitive style of agents, the LRGA model is grounded in theories in psychology and sociology. However, it focuses on other main factors in team activity, like synchronization and convergence [1,22,23,27]. Hence, including knowledge representation and agent memory in the proposed model was important. Many MAS consider only simple memory descriptions, like weighted averages of previously visited solutions in KABOOM [34]. Instead, each LRGA agent has a local memory that stores the previous experience and newly learned ideas generated by the agent or received from other agents. A Bayesian framework was devised for idea understanding in the LRGA model to identify the more likely meanings of inputs, while traditional MAS models do not emphasize input ambiguity as it is less common for mathematical or logic expressions. Our experiments show that ambiguity can be reduced during agent interaction by responses with higher or lower utilities, which favor reaching a common understanding, while alternatives with similar utilities increase ambiguity. We argue that idea disambiguation requires more powerful knowledge and memory representations than those based on aggregated measures, like weighted averages.

New ideas are generated in the LRGA model using either a fast, top-down flow or a slower, bottom-up flow guided by parameters grounded in psychology, like affective valuation, to determine the nature of emotions based on the differences between expected outcomes and received inputs [16,24,49], responses to asymmetrical utilities [58], dominant concepts that capture the agent's attention [38], minimizing the required effort [58], or maximizing the accessed knowledge. Idea creation through the slower, bottom-up flow is similar to the five design strategies discussed in ref. [50,51,59], and is also supported by the model presented in ref. [17]. Our experiments show that reaching consensus among agents is achieved by strategies specific to different scenarios, such as strategies that tackle response utility and response structure and length in a certain way. These strategies allow a more flexible switching between incremental changes, like through the fast flow, and more significant changes, such as through the slow flow, than other MAS, like KABOOM, where the switching between incremental and significant changes mainly depends on the time since the start of problem solving, i.e., significant changes are early and incremental changes are later in the process. However, studies show that reaching better solutions, including departures from previous solutions and breakthroughs, does not happen early but later during solving [2,7]. Furthermore, our strategy selection is different from the model in ref. [36], where randomly selected features propagate more likely for the more similar agents without considering the utility of the features.

7. Conclusions

This paper proposes a novel agent-based model, called Learning and Response Generating Agents (LRGAs), to study the conditions under which team problem solving ends either in consensus or disagreement between agents. Agents interact with each other through structured messages that include nouns, verbs, adjectives, and adverbs. An LRGA learns by updating its own memory based on the received inputs and the responses that it creates. Each agent understands an input by matching it to the most similar statements in its own memory to clarify possible ambiguities in the input. Furthermore, an LRGA creates new responses by combining concepts through strategies selected based on the differences in utility between the input and the matched memory item and emotional and social cues. The available strategies can explore variants of a response by adding more details or considering alternatives of similar meaning, establishing abstractions for a set of similar responses, picking up less frequent concepts, and reasoning with more abstract responses by combining their structures. The difference between the utility of an expected outcome and that of a received input selects between a fast, top-down and a slow, bottom-up version of each strategy. The LRGA model was devised to study problems that require creating new ideas accepted by all team members, like problem framing and common ground finding, and to support devising new Human-Computer Interfaces for self-improvement and teaching.

Experiments showed that the generated responses pertain to ten possible situations depending on the selected strategies for the fast or slow flows: input learning, reinforcing an existing experience item, creating new instances of inputs or experience items, creating synonyms of inputs or experience items, creating homonyms of inputs or experience items, and creating abstractions of inputs or experience items. Clause ambiguity is reduced by inputs of different utilities than the utilities previously associated by an agent to the similar memory item, as the differences encourage the agent either to adopt or separate from the input, thereby lowering ambiguity. Inputs of similar utility tend to increase ambiguity. Fast same utility and fast same utility with alternatives, as well as slow higher utility strategies, offer the best connectivity improvement for already linked concepts because they are more likely to relate the nouns that do not describe an objective. Fast lower utility and slow same utility methods improve the connectivity of unlinked concepts because they tend to connect the nouns expressing an objective. Longer responses also increase the likelihood of increasing the connectivity of concepts. All these cases help consensus reaching. Disagreement increases through the fast higher utility strategy as the created responses are based on the differences between input and memory, thus enlarging the information gap between agents.

The main limitations of the LRGA model relate to the fact that language statements are used for new idea generation, communication, and understanding by the agents. This raises significant challenges in semantic understanding, not only for language constructs like analogies and metaphors but also for more abstract or ambiguous situations that can be understood only based on the current context and experience. Agents currently have only a small memory to store experience and context; hence, they can handle only simple disambiguation. Another limitation relates to tackling abstractions and negations. The model defines abstractions as placeholders for a set of similar concepts, but there is no mechanism to support attribute aggregation. Negation focuses only on the opposite utility of an idea or concept but does not consider how the causes of negation can be the starting point for creating a new solution. More strategies to generate responses are likely needed.

Future work will continue the study of other interaction mechanisms in the LRGA model and define metrics to describe the dynamics of team behavior over time. We also intend to extend the method for response understanding by incorporating more insight from linguistics, such as that presented in ref. [17]. We think that improving the language part of the agents is the starting point in any effort to computationally model with reasonable accuracy the behavior of human teams participating in problem solving.

Author Contributions: Conceptualization, A.D.; methodology, A.D. and D.-I.C.; validation, A.D. and D.-I.C.; formal analysis, A.D. and D.-I.C.; investigation, A.D. and D.-I.C.; resources, A.D.; data curation, A.D.; writing—original draft preparation, A.D. and D.-I.C.; writing—review and editing, A.D. and D.-I.C.; supervision, A.D. All authors have read and agreed to the published version of th manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Brennan, S.; Galati, A.; Kuhlen, A. Two Minds, One Dialog: Coordinating Speaking and Understanding. In *The Psychology of Learning and Motivation: Advances in Research and Theory*; Ross, B.H., Ed.; Volume 53: Psychology of Learning and Motivation; Academic Press: Cambridge, MA, USA, 2010; pp. 301–344. [CrossRef]
2. Dorst, K. The core of 'design thinking' and its application. *Des. Stud.* **2011**, *32*, 521–532. [CrossRef]
3. Suk, K.; Lee, H. Problem Framing Activities Carried Out by Student Design Teams to Enhance Creativity: Comparative Analysis of High and Low Creative Teams. *Arch. Des. Res.* **2021**, *34*, 23–38. [CrossRef]
4. Coursey, M.; Williams, B.; Kenworthy, J.; Paulus, P.; Doboli, S. Divergent and Convergent Group Creativity in an Asynchronous Online Environment. *J. Creat. Behav.* **2020**, *54*, 253–266. [CrossRef]
5. Doboli, A.; Liu, X.; Li, H.; Doboli, S. Modeling Group Creativity as the Evolution of Community-level, Creative Problem Solving. In *The Oxford Handbook*; Oxford University Press: Oxford, UK, 2019.
6. Doboli, A.; Doboli, S. A novel agent-based, evolutionary model for expressing the dynamics of creative open-problem solving in small groups. *Appl. Intell.* **2021**, *51*, 2094–2127. [CrossRef] [PubMed]
7. Umbarkar, A.; Subramanian, V.; Doboli, A.; Doboli, S. Two Experimental Studies on Creative Concept Combinations in Modular Design of Electronic Embedded Systems. *Des. Stud.* **2014**, *35*, 80–109. [CrossRef]
8. Tomasello, M. *Origins of Human Communication*; MIT Press: Cambridge, MA, USA, 2010.
9. Malone, T.; Bernstein, M. *Handbook of Collective Intelligence*; The MIT Press: Cambridge, MA, USA, 2015.
10. Abney, D.; Paxton, A.; Dale, R.; Kello, C. Complexity Matching in Dyadic Conversation. *J. Exp. Psychol. Gen.* **2014**, *143*, 2304–2315. [CrossRef] [PubMed]
11. Smith, E.; Conrey, R. Agent-Based Modeling: A New Approach for Theory Building in Social Psychology. *Personal. Soc. Psychol. Rev.* **2007**, *11*, 87–104. [CrossRef] [PubMed]
12. Furini, M.; Gaggi, O.; Mirri, S.; Montangero, M.; Pelle, E.; Poggi, F.; Prandi, C. Digital Twins and Artificial Intelligence: As Pillars of Personalized Learning Models. *Commun. ACM* **2022**, *65*, 98–104. [CrossRef]
13. Doboli, A.; Doboli, S.; Duke, R.; Hong, S.; Tang, W. Dynamic Diagnosis of the Progress and Shortcomings of Student Learning using Machine Learning based on Cognitive, Social, and Emotional Features. *arXiv* **2022**, arXiv:2204.13989. https://doi.org/10.48550/ARXIV.2204.13989.
14. Chaouche, A.; Fallah Seghrouchni, A.; Ilié, J.-M.; Saidouni, S. A Higher-order Agent Model for Ambient Systems. *Procedia Comput. Sci.* **2013**, *21*, 156–163. [CrossRef]
15. Cacioppo, J.T.; Visser, P.S.; Pickett, C.L. (Eds.) *Social Neuroscience: People Thinking about Thinking People*; MIT Press: Cambridge, MA, USA, 2006.
16. Lowe, R.; Almer, A.; Balkenius, C. Bridging Connectionism and Relational Cognition through Bi-directional Affective-Associative Processing. *Open Inf. Sci.* **2019**, *3*, 235–260. [CrossRef]
17. Fauconnier, G.; Turner, M. *The Way We Think. Conceptual Blending and The Mind'S Hidden Complexities*; Basic Books: New York, NY, USA, 2002.
18. Pickering, M.; Garrod, S. Toward a mechanistic psychology of dialogue. *Behav. Brain Sci.* **2004**, *27*, 169–190. [CrossRef] [PubMed]
19. Shockley, K.; Richardson, D.; Dale, R. Conversation and Coordinative Structures. *Top. Cogn. Sci.* **2009**, *1*, 305–319. [CrossRef] [PubMed]
20. Costello, F.J.; Keane, M.T. Testing two theories of conceptual combination: Alignment versus diagnosticity in the comprehension and production of combined concepts. *J. Exp. Psychol. Learn. Mem. Cogn.* **2001**, *27*, 255–271. [CrossRef]
21. Fellbaum, C. *WordNet. An Electronic Lexical Database*; The MIT Press: Cambridge, MA, USA, 1998.
22. Vesper, C.; Sangati, E.; Butepage, J.; Ciardo, F.; Crossey, B.; Effenberg, A.; Hristova, D.; Karlinsky, A.; McEllin, L.; Nijssen, S.; et al. Joint Action: Mental Representations, Shared Information and General Mechanisms for Coordinating with Others. *Front. Psychol.* **2016**, *7*, 2039. [CrossRef]
23. Fusaroli, R.; Ryczaszek-Leonardi, J.; Tylén, K. Dialog as interpersonal synergy. *New Ideas Psychol.* **2014**, *32*, 147–157. [CrossRef]
24. Lowe, R.; Almer, A.; Lindblad, G.; Gander, P.; Michael, J.; Vesper, C. Minimalist Social-Affective Value for Use in Joint Action: A Neural-Computational Hypothesis. *Front. Comput. Neurosci.* **2016**, *10*, 88. [CrossRef]
25. Tanenhaus, M.; Spivey-Knowlton, M.; Eberhard, K.; Sedivy, J. Integration of visual and linguistic information in spoken language comprehension. *Science* **1995**, *268*, 1632–1634. [CrossRef]
26. Fowler, C.; Richardson, M.; Marsh, K.; Shockley, K. *Language Use, Coordination, and the Emergence of Cooperative Action*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 2008, pp. 261–279. [CrossRef]
27. Horton, W.; Keysar, B. When do speakers take into account common ground? *Cognition* **1996**, *59*, 91–117. [CrossRef]
28. Keysar, B.; Barr, D.; Balin, J.; Brauner, J. Taking Perspective in Conversation: The Role of Mutual Knowledge in Comprehension. *Psychol. Sci.* **2000**, *11*, 32–38. [CrossRef]
29. Axelrod, R. *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*; Princeton University Press: Princeton, NJ, USA, 1997.
30. Baldoni, M.; Baroglio, C.; Capuzzimati, F.; Micalizio, R. Commitment-based Agent Interaction in JaCaMo$^+$. *Fundam. Inform.* **2018**, *159*, 1–33. [CrossRef]
31. Chopra, A.; Artikis, A.; Bentahar, J.; Dignum, F. Introduction to the special section on agent communication. *ACM Trans. Intell. Syst. Technol.* **2013**, *4*, 1. [CrossRef]

32. Halpern, J.; Shore, R. Reasoning About Common Knowledge with Infinitely Many Agents. *arXiv* **1999**, arXiv:cs/9909014. [CrossRef].
33. Hubner, J.; Sichman, J. $S - Moise^{+}$: A Middleware for Developing Organised Multi-agent Systems. In Proceedings of the International Conference on Agents, Norms and Institutions for Regulated Multi-Agent Systems, Utrecht, The Netherlands, 25–26 July 2005; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3913, pp. 64–77. [CrossRef]
34. Lapp, S.; Jablokow, K.; McComb, C. KABOOM: An agent-based model for simulating cognitive style in team problem solving. *Des. Sci.* **2019**, *5*, e13. [CrossRef]
35. Lapp, S.; Jablokow, K.; McComb, C. Collaborating with style: Using an agent-based model to simulate cognitive style diversity in problem solving teams. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Anaheim, CA, USA, 18–21 August 2019; American Society of Mechanical Engineers: New York, NY, USA, 2019; Volume 59278, p. V007T06A029. [CrossRef]
36. Axelrod, R. The Dissemination of Culture: A Model with Local Convergence and Global Polarization. *J. Confl. Resolut.* **1997**, *41*, 203–226. [CrossRef]
37. Iqbal, S.; Sha, F. Actor-Attention-Critic for Multi-Agent Reinforcement Learning. *arXiv* **2019**, arXiv:1810.02912. [CrossRef].
38. Liu, Y.; Wang, W.; Hu, Y.; Hao, J.; Chen, X.; Gao, Y. Multi-agent game abstraction via graph attention neural network. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 7211–7218. [CrossRef]
39. Zhang, S.; Zhang, Q.; Lin, J. Efficient Communication in Multi-Agent Reinforcement Learning via Variance Based Control. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.; Garnett, R., Eds.; Curran Associates: New York, NY, USA, 2019; Volume 32.
40. Das, A.; Gervet, T.; Romoff, J.; Batra, D.; Parikh, D.; Rabbat, M.; Pineau, J. Tarmac: Targeted multi-agent communication. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 10–15 June 2019; pp. 1538–1546.
41. Chopra, A.; Artikis, A.; Bentahar, J.; Colombetti, M.; Dignum, F.; Fornara, N.; Jones, A.; Singh, M.; Yolum, P. Research directions in agent communication. *ACM Trans. Intell. Syst. Technol.* **2013**, *4*, 1–23. [CrossRef]
42. Telang, P.; Singh, M.; Yorke-Smith, N. Maintenance of Social Commitments in Multiagent Systems. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; AAAI Press: Cambridge, MA, USA, 2021; pp. 11369–11377. [CrossRef]
43. Singh, M.; Chopra., A. Clouseau: Generating Communication Protocols from Commitments. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; AAAI Press: Cambridge, MA, USA, 2020; pp. 7244–7252. [CrossRef]
44. Bordini, R.; Hubner, J.; Tralamazza, D. Using *Jason* to Implement a Team of Gold Miners. In Proceedings of the International Conference on Computational Logic in Multi-Agent Systems, Hakodate, Japan, 8–9 May 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 304–313. [CrossRef]
45. Doboli, A.; Doboli, S. A Novel Learning and Response Generating Agent-based Model for Symbolic—Numeric Knowledge Modeling and Combination. In Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 5–7 December 2021. [CrossRef]
46. Doboli, A.; Umbarkar, A.; Doboli, S.; Betz, J. Modeling semantic knowledge structures for creative problem solving: Studies on expressing concepts, categories, associations, goals and context. *Knowl.-Based Syst.* **2015**, *78*, 34–50. [CrossRef]
47. Ferent, C.; Doboli, A.; Doboli, S. An axiomatic model for concept structure description and its application to circuit design. *Knowl.-Based Syst.* **2013**, *45*, 114–133. [CrossRef]
48. Ellsworth, P.; Scherer, K. Appraisal processes in emotion. In *Handbook of the Affective Sciences*; Davidson, R.J., Goldsmith, H.H., Scherer, K.R., Eds.; Oxford University Press: New York, NY, USA, 2003; pp. 572–595.
49. Ortony, A.; Clore, G.; Collins, A. *The Cognitive Structure of Emotions*; Cambridge University Press: Cambridg, UK, 1988.
50. Jiao, F.; Montano, S.; Ferent, C.; Doboli, A.; Doboli, S. Analog Circuit Design Knowledge Mining: Discovering Topological Similarities and Uncovering Design Reasoning Strategies. *IEEE Trans. CADICS* **2015**, *34*, 1045–1059. [CrossRef]
51. Jiao, F.; Doboli, A. Knowledge-Intensive, Causal Reasoning for Analog Circuit Topology Synthesis in Emergent and Innovative Applications. In Proceedings of the Design, Automation and Test in Europe Conference (DATE), Grenoble, France, 9–13 March 2015; pp. 1144–1149.
52. Barsalou, L. Perceptual symbol systems. *Behav. Brain Sci.* **1999**, *22*, 577–660. [CrossRef] [PubMed]
53. Barsalou, L. Abstraction in perceptual symbol systems. *Philos. Trans. R. Soc. Lond. B* **2003**, *358*, 1177–1187. [CrossRef] [PubMed]
54. Duke, R.; Doboli, A. Top-down Approach to Solving Speaker Diarization Errors in diaLogic System. In Proceedings of the IEEE International Symposium on Smart Electronic Systems (iSES), Warangal, India, 18–22 December 2022; pp. 213–218. [CrossRef]
55. Deco, G.; Rolls, E. Attention, short-term memory, and action selection: A unifying theory. *Prog. Neurobiol.* **2005**, *76*, 236–256. [CrossRef] [PubMed]
56. Spitzer, H.; Desimone, R.; Moran, J. Increased attention enhances both behavioral and neuronal performance. *Science* **1988**, *240*, 338–340. [CrossRef]
57. Parkinson, B. What holds emotions together? Meaning and response coordination. *Cogn. Syst. Res.* **2009**, *10*, 31–47. [CrossRef]
58. Gigerenzer, G.; Todd, P.; Group, A.R. *Simple Heuristics That Make Us Smart*; Oxford University Press: Oxford, UK, 1999.

59. Li, H.; Liu, X.; Doboli, A.; Doboli, S. InnovA: A Cognitive Architecture for Computational Innovation through Robust Divergence and Its Application for Analog Circuit Design. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2018**, *37*, 1943–1956. [CrossRef]