

**THE ROLE OF TRUST AND RELATIONSHIPS IN HUMAN-ROBOT
SOCIAL INTERACTION**

A Dissertation
Presented to
The Academic Faculty

by

Alan Richard Wagner

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Computer Science

Georgia Institute of Technology
December, 2009

COPYRIGHT 2009 BY ALAN R WAGNER

**THE ROLE OF TRUST AND RELATIONSHIPS IN HUMAN-ROBOT
SOCIAL INTERACTION**

Approved by:

Dr. Ronald C. Arkin, Advisor
College of Computing
Georgia Institute of Technology

Dr. Henrik I. Christensen
College of Computing
Georgia Institute of Technology

Dr. Ashwin Ram
College of Computing
Georgia Institute of Technology

Dr. Andrea Tomaz
College of Computing
Georgia Institute of Technology

Dr. Arthur D. Fisk
School of Psychology
Georgia Institute of Technology

Date Approved: October, 26 2009

*To my family and friends,
without your love and support
this would never have been possible*

“This above all: to thine own self be true,
And it must follow, as the night the day,
Thou canst not then be false to any man.”

—William Shakespeare

“I have as much authority as the Pope, I just don't have as many people who believe it.”

—George Carlin

ACKNOWLEDGEMENTS

The completion of a dissertation is, in some ways, a collective effort necessitating the support of friends, family, and mentors. I would like to thank all of these people, whom in their own various ways, made this effort possible.

I would like to thank my advisor, Dr. Ron Arkin. The kernel from which this dissertation grew came from a brief discussion that Ron and I had. Ron stated, “How do I know if I should trust an (robotic) advisor?” I replied, “Perhaps my topic should be trust and advisement?” He replied, “That sounds good.” And the thesis topic was born. With time it has evolved into the current document. Ron’s support has been a consistent and critical reason for the completion of this work.

I would like to thank my family for their love, encouragement and support during these years. In their own way, “the girls,” Allaina and Gabe, have helped me remember why I am here. Cora, your love has often reminded me of just how important relationships are. To Colleen, thank you for your long and patient support.

I would also like to recognize the support of my mother and father, who, by asking when I will be finished have reminded me that I need to finish. My mother completed her bachelor’s degree around the same time that I completed mine. Her hard work has always been an important inspiration to me. My father’s natural genius and leadership has also inspired me. Self taught and always able to fix, construct, or repair anything, his intuitive genius has oft amazed me. Always a leader when interacting with others, I have tried to emulate him in many ways. Moreover, his tidbits of wisdom, such as “make sure your job is something that you love” have influenced me from childhood.

To my brother Robert and my sister Danielle, there will soon be a doctor in the family. To my grandfather Richard, although you died just before I was born I know you would have been proud. To Pop, your op-ed writing has been an inspiration for my own writing. To Mema, we all miss you. Although I never became a medical doctor, I think you would have been nearly as proud of a computer scientist. And to grandma, your life, your strength has always and continues to be an influence in my life.

I would also like to thank my dissertation committee, Andrea Tomaz, Henrik Christensen, Dan Fisk, and Ashwin Ram. Thank you for your encouragement and direction with these topics.

I would like to recognize the researchers and interns of the Naval Research Laboratory, such as Greg Trafton, Ben Frasen, Wende Frost, and Alan Schultz. NRL is an exciting and fun place to research critical topics.

Finally, I would like to thank my many many friends who have made this possible in the beginning and in the end. To Zsolt Kira and Patrick Ulam, friends from the first day of graduate school, with your support this would not have been possible. Yoichiro Endo, always a leader in the lab and a source of knowledge, thank you for all of your help. I would also like to recognize the other students that made the lab a rich and exciting place to be: Brian Lee, Keith O'Hara, Ananth Ranganathan, Brittany Duncan, and Alexander Stoytchev. Last, but certainly not least, I would like to recognize the immense support of Doug and Shawna Judson, Debbie and Alan Harkin, and Amy Ramsey for all of there encouragement in the end.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	v
LIST OF TABLES	x
LIST OF FIGURES	xiv
LIST OF ALGORITHMS	xxiv
LIST OF SYMBOLS AND ABBREVIATIONS	xxvi
SUMMARY	xxviii
 <u>CHAPTER</u>	
1 INTRODUCTION	1
1.1 Motivation	3
1.2 Principal Research Question	5
1.3 Objectives	7
1.4 Dissertation Outline	8
 2 CHARACTERIZING HUMAN-ROBOT SOCIAL RELATIONS: A REVIEW	 9
2.1 Human-Robot Interaction	10
2.1.1 Interactive communication	10
2.1.2 Styles of interaction	11
2.1.3 HRI methodology	12
2.2 Interpersonal Relations	14
2.2.1 Relationship theory	15
2.2.2 Relationship and social situation analysis	23
2.2.3 Social learning	25
2.2.4 Connections to game theory	28
2.3 Using Trust to Characterize Relationships	29
2.3.1 Definitions of trust	29
2.3.2 Using social situations to evaluate trust	32
2.3.3 Alternative methods for evaluating trust	34
2.4 Summary	40
 3 A METHODOLOGY FOR INVESTIGATING THE THEORY UNDERLYING HUMAN-ROBOT INTERACTION	 42
3.1 A Method for HRI Research	42
3.2 Controlling Human Behavior—Actor Scripts	45
3.3 Controlling the Environment—Social Situations	46
3.4 Evaluative Methods	48
3.4.1 Numerical simulation experiments	48
3.4.2 Simulation experiments within a simulation environment	50
3.4.3 Laboratory experiments	54
3.4.4 Evaluation platforms	55
3.4.5 Interactive communication	56

3.4.6	Human partners	57
3.4.7	Perceptual underpinnings	58
3.4.8	Robot behaviors	59
3.5	Example Interaction	60
4	A FRAMEWORK FOR REPRESENTING AND REASONING ABOUT HUMAN ROBOT SOCIAL INTERACTION.....	62
4.1	Defining the term Social Interaction.....	62
4.1.1	Social situations and interaction	64
4.1.2	A formal notation for describing human-robot interaction.....	65
4.2	Partner Modeling	68
4.3	The Transformation Process	74
4.3.1	Transforming an outcome matrix.....	76
5	FROM PERCEPTION TO OUTCOME MATRIX	79
5.1	Developing an Algorithm for Outcome Matrix Creation	79
5.2	Outcome Matrix Error Sensitivity	83
5.2.1	Errors in outcome value magnitude	88
5.2.2	Errors in individual outcome	90
5.2.3	Action deletion errors	94
5.2.5	Action insertion errors	97
5.2.6	Error sensitivity conclusions.....	99
5.3	The Interact and Update Algorithm	100
5.3.1	Creating accurate partner models.....	106
5.3.2	Interact and update conclusions	117
5.4	The Stereotyping Matching Algorithm.....	119
5.4.1	Examining the use of stereotypes	125
5.4.2	Stereotype matching conclusions.....	134
5.5	Creating Outcome Matrices: Conclusions	136
6	SITUATION ANALYSIS	138
6.1	Situation-based Human-Robot Social Interaction	139
6.1.1	Interdependence theory	140
6.1.2	The situation analysis algorithm	142
6.1.3	Using situation analysis to select interactive behaviors.....	146
6.1.4	Mapping a situation's location to a transformation	148
6.1.5	A computational process for situation analysis	151
6.2	Experiments and Result	154
6.2.1	Situation analysis in practice	154
6.2.2	Situation analysis over the entire interdependence space	161
6.3	Situation Analysis Conclusions	165
7	REPRESENTING AND REASONING ABOUT RELATIONSHIPS.....	168
7.1	Relational Disposition.....	169
7.2	Diagnostic Situations	173
7.2.1	Diagnostic Situations as a function of Matrix Size.....	184

7.3	Characterizing Relationships	188
7.3.1	Accuracy of Relational Disposition Algorithm	190
7.4	Conclusions.....	196
8	TRUST IN HUMAN-ROBOT INTERACTIONS.....	197
8.1	Recognizing Situations that Require Trust	198
8.1.1	Interdependence space mapping of situational trust	203
8.1.2	Canonical situations and the conditions for trust.....	207
8.2	Measuring Trust	212
8.3	Recognizing Relationships that afford Trust	216
8.3.1	Selecting a Trusted Partner	217
8.3.2	Selecting the most Trusted Partner	219
8.4	Conclusions: Trust in Human-Robot Interactions	228
9	CONCLUSIONS.....	229
9.1	Summary of Contributions.....	229
9.2	Research Questions Revisited.....	231
9.3	The Road Ahead	233
9.4	Towards a Socially Intelligent Robot	236
APPENDIX A:	GLOSSARY OF TERMS	237
APPENDIX B:	EXAMPLE SOCIAL SITUATIONS.....	240
APPENDIX C:	LIST OF TRANSFORMATION TYPES	242
REFERENCES:	243

LIST OF TABLES

	Page
Table 2.1: The dimensions of interdependence space are listed with descriptions of the maximal and minimal values and examples at the extremes.	20
Table 2.2: A list of the various measures and models of trust used in previous research. The meaning of the symbols are presented within the text of this section.	39
Table 3.1: List of colored objects in each environment.	51
Table 3.2: A list of partner features is presented above. Several of the features were devised because of their notional significance.....	57
Table 3.3: A summary of the perceptual requirements, the software package used, and their usage.	58
Table 3.4: The table provides a summary of the behaviors used.	60
Table 4.1: The table provides a list of some example transformations. The table provides the name of the transformation, a description, the computational arrangement of the transformation, and the character displayed by an individual that often selects the type of transformation. A more complete list is provided in Appendix C.	77
Table 5.1: Environment types, partner types, and robot types for the outcome matrix error sensitivity experiment.	84
Table 5.2: Example actions for different types of individuals.	85
Table 5.3: Experimental procedure for the outcome matrix error sensitivity experiments.	87
Table 5.4: Experiment summary for the errors in outcome value magnitude experiment.....	89
Table 5.5: Experiment summary for the errors in individual outcome values experiment.....	91
Table 5.6: Experiment summary for the action deletion experiment.	94
Table 5.7: Experiment summary for the action insertion errors experiment.	97

Table 5.8: General experimental information common to all of the experiments performed to investigate the use of the interact-and-update algorithm for the creation of accurate partner models.	106
Table 5.9: Summary of the creating accurate partner models experiment conducted in simulation with a single partner type and in multiple environments.....	107
Table 5.10: List of actions available to the robot for each different type of environment.	108
Table 5.11: A list of actions for each type of partner.....	108
Table 5.12: Procedure for creating partner models.....	109
Table 5.13: Experimental procedure used for each of the experiments in this section.	109
Table 5.14: Summary of the creating accurate partner models experiment conducted in simulation with multiple partners and in a single environment.....	112
Table 5.15: Summary of the creating accurate partner models experiment conducted in simulation with multiple partners and in a single environment.....	115
Table 5.16: Summary of the use of stereotypes experiment conducted in simulation with multiple partners and in a single environment.....	125
Table 5.17: Experimental procedure used in the examining the use of stereotypes simulation experiment. The experiment compares an experimental condition to a control condition. Steps 2 and 4, therefore, only occur in the experimental condition.....	128
Table 5.18: This table depicts the change in number of clusters and decision tree structure as the robot progressively interacted with different partners during the experiment. We see that by the seventh partner the robot as created clusters for each type. Moreover, after interacting with this seventh partner the robot's decision tree accurately assigns a stereotype model based on the partner's perceptual features (Figure 5.12).	130
Table 5.19: Experimental summary for the laboratory experiment relating to the use of stereotypes.	132
Table 6.1: Calculation of the interdependence space dimensions given the variances from Figure 6.3. Equations (4) and (5) are from (Kelley & Thibaut, 1978), (6) and (7) were developed by the author.....	145
Table 6.2: A list of several simple matrix transformations. The list is not exhaustive.	147

Table 6.3:	The cells denote the mean outcome obtained by the transformation at each location. The shaded cells indicate the mean of the best transformation. The confidence interval is included for all values.....	150
Table 6.4	Experimental summary for the situation analysis experiment conducted in a search and rescue environment.	155
Table 6.5	Experimental summary for the situation analysis experiment conducted in a search and rescue environment.	162
Table 7.1	A list of several different types of transformations and a description of each. A relational disposition describes an individual’s tendency to use a single transformation when interacting with a particular partner. Hence, the table below describes several relational dispositions.....	173
Table 7.2	The table below lists the diagnostic characteristics for different combinations of robot transformation type, first transformation type, second transformation type, and comparator. Each of these combinations does not result in an inverted characterization. The combinations of robot type, hypothesized transformation type and method of comparison only result in diagnostic and non-diagnostic situations and, hence, can be used to determine the partner’s relational disposition.....	178
Table 7.3	Summary of the diagnostic situations as a function of matrix size experiment.....	185
Table 7.4	Experimental procedure for the diagnostic situations as a function of matrix size experiment.....	186
Table 7.5	Summary of the relational disposition algorithm experiment.....	191
Table 7.6	Experimental procedure for the relational disposition algorithm experiment.....	193
Table 8.1	Summary of the interdependence space mapping of situational trust experiment.....	203
Table 8.2	Experimental procedure for the interdependence space mapping of situational trust experiment.....	204
Table 8.3	Summary of the canonical situations and the conditions for trust experiment.....	207

Table 8.4 Several situations arbitrary situations are depicted above. The table includes a description of the situation and the situation's outcome matrix. The first condition the algorithm in Box 8.1 is assumed to hold for all situations. Columns 3-5 present the results for the remaining conditions. The right most column presents the algorithms final output, stating whether or not the situation demands trust.209

Table 8.5 The table demonstrates the change in trust measure with respect to the changing conditions of the investor-trustee game. The table shows that as the probability that the trustee will violate the trust increases, so to does the trust measure. Hence the amount of trust necessary to selected the trusting action increases. Moreover, as the loss increases in relation to the initial money given to the investor the trust measure increases.....214

Table 8.6 Summary of the selecting the most trusted partner experiment.....219

Table 8.7 Experimental procedure for the most trusted partner experiment.224

LIST OF FIGURES

	Page
Figure 2.1: A process model for turn-taking interaction is depicted above. This style of interaction involves iterative distinct responses.	11
Figure 2.2: Although extremely mobile, robots such as the Clodbusters (Hsieh et al., 2007) on the left have been designed with little capacity for interaction. Kismet to the right, on the other hand, has been designed specifically for interaction. Although many degrees of freedom control its facial expressions the robot was not designed for general purpose motion over long distances	13
Figure 2.3: The distribution to the left depicts a two dimensional cross-section of the interdependence space. Here some prototypical relationships are described in terms of their interdependence and correspondence. To the right, a three dimensional distribution adds the basis of control dimension. This cube depicts prototypical interpersonal situations such as the prisoner's dilemma game (PDG) and trust game as squares within the cube. These prototypical situations often occur on planes within the cube (adapted from Kelley, 1979 and Kelley et al., 2003).....	18
Figure 2.4: An example outcome matrix is depicted. The term $_{xy}o^1$ denotes the first individual's outcomes and the terms $_{xy}o^2$ denote the second individual's outcomes. Outcomes result from the selection of an action pair by each individual.....	20
Figure 2.5: The transformation process (adapted from Rusbult & Van Lange, 2003). This process model transforms a given or perceived situation into behavioral selection by a person. The given situation is influenced by cognitive factors such as relational motives and emotions. These factors transform the perceived situation into an effective situation that the individual uses for selecting social behavior.	21
Figure 2.6: Example outcome matrices are depicted above. The original outcome matrix is on far left. These outcomes are converted into the BAC, MPC, and MJC by following the procedure listed at the top of the figure. The numbers for the raw outcomes are provided in the example parameters listed to the left. In this example the robot selects between rescue and cleanup actions.	24

Figure 3.1: Screenshots from the simulation environment are depicted above. The top left shows a household environment. The top right depicts a museum environment. The bottom left illustrates the assistive environment. The bottom right illustrates the prison environment.	51
Figure 3.2: The interface used by the human to move and interact in the simulated environment. The environment shown is the search and rescue environment.....	52
Figure 3.3: Depiction of the network and control setup used to perform simulation experiments. The human interacts through the Human Control Interface. The simulation server runs the simulation environment and feeds information to both the Human Control Interface and the Robot Client. ...	53
Figure 3.4: The figure depicts a split screen view of the search and rescue environment. The human sees only the top half of the figure. The bottom half shows the robot situated in the environment.....	54
Figure 3.5: An overview of the maze environment used as a mockup of a simple search and rescue environment. One corner of the maze had two dolls representing children (center photo) and the other corner had a simulated fire and biohazard (right photo).	54
Figure 3.6: A photo of the Pioneer DX used to perform the laboratory experiments. ..	55
Figure 3.7: A photo of the Rovio mobile robot. The robot's neck is point towards the bottom of the image and is in the unextended position. The webcam is at the end of the neck.	56
Figure 3.8: In laboratory experiments the robot uses landmark detection to aid navigation.	59
Figure 4.1: This figure depicts the difference between an interaction and a social situation. Both outcome matrices above depict the Prisoner's dilemma. The left most matrix depicts the prisoner's dilemma as an interaction between two specific people selecting between specific actions. The right most matrix depicts the prisoner's dilemma as an abstract social situation, without specific actions or individuals.	65
Figure 4.2: The figure above demonstrates the use of our outcome matrix notation in a human-robot interaction experiment conducted by German Research Center for Artificial Intelligence (Zender, Mozos, & Jensfelt, 2007). The robot (pictured in the top right photo) asks the human whether or not a door is present. The human says no 10 times before the robot responds. Notation is provided for the robot and the human. The human's outcomes decrease every time he must repeat the command. Finally, when the robot responds, the human's outcomes are increase dramatically.....	69

Figure 4.3: The figure above provides an example measurement of partner model accuracy. The robot currently has an action model for partner consisting of two actions, one of which is not correct. The same is true of the robot's utility function for the partner. Calculations are provided in the lower half of the diagram. The resulting distance from the true model is $d=0.5$ 71

Figure 4.4: The figure above depicts our framework for social action selection. Situation features are used to generate the given situation (described in section 5.3). The given situation is transformed to include the robot's disposition producing the effective situation. Finally, an action is selected from the effective situation. 75

Figure 5.1: An example of the Outcome Matrix Creation algorithm in a search and rescue environment with a firefighter. Step 1 begins with an empty matrix which is filled with information related to the interaction. The result is the final matrix labeled Steps 6-8. 82

Figure 5.2: Diagram depicting the process used to create the partner models for the four error sensitivity experiments presented in the following four subsections. 88

Figure 5.3: An example outcome matrix from the error sensitivity experiments. The robot's use of a *max_own* strategy would result in selection of the *alert-fire* action. 90

Figure 5.4: The graph depicts the percent of incorrect actions selected as a function of errors in outcome values. A y-axis value of 1.00 represents total selection of incorrect actions. The bold black line depicts the average incorrect actions selected for all environments. The individual colored lines represent changes in accuracy for each different environment. The bold white line is a baseline for comparison, depicting a linear decrease in accuracy. The fact that the bold black line is below the bold white line indicates that errors in outcome value result in less than linear action selection error. 93

Figure 5.5: The graph depicts the percent of incorrect actions selected as a function of increasing random action deletion. The bold black line depicts the average incorrect actions selected for all environments. The individual colored lines represent changes in accuracy for each different environment. The bold white line is a baseline for comparison, depicting a linear decrease in accuracy. Note that the black line approximates the white line. Hence, in contrast to the two previous experiments this type of error increases approximately linearly. 96

Figure 5.6: The graph depicts the percent of incorrect actions selected as a function of increasing random action insertion. As in the other graphs the bold black line depicts the average incorrect actions selected for all environments, the colored lines represent the changes in accuracy for each different environment, and the bold white line is a baseline for comparison, depicting a linear decrease in accuracy. 99

Figure 5.7: This figure presents an example run through the interact-and-update algorithm. The partner and situation features are presented as inputs to the algorithm. In steps 1 and 2 these features are used to retrieve the partner and self models. In step 3, the models are used to create the pictured matrix. Steps 4 and predict actions and outcomes based on the models. Step 6 performs the action and steps 7 through 12 update the models. Steps 13 and 14 delete unused actions, if necessary and Step 15 goes back to Step 3.102

Figure 5.8: The graph depicts the results from the first simulation experiment involving different environments. The results show that model accuracy increases with continued interaction, eventually matching the target model.....110

Figure 5.9: The graph depicts the results from the second simulation experiment involving different partner types. The results again show that model accuracy increases with continued interaction, eventually matching the target model.114

Figure 5.10: Photos from the robot experiment. The robot initially moves to observe the victim. After learning the model of its partner the robot moves to observe the hazard. Photos 1-4 depict the robot as it moves through the maze and selects actions. Photo 5 depicts video that the robot sends to its human partner.116

Figure 5.11: An example run of the stereotype building algorithm. Three partner models serve as input to algorithm. In the cluster phase, the algorithm first merges models M1 and M2 creating model M12. The distance between model M3 and M12 is greater than 0.25 so the model is not merged into the stereotype. In the function learning phase the stereotype clusters are paired with the partner features. A classifier is constructed from the resulting data. The example skips some of the most easily understood steps, such as the for loops.....123

- Figure 5.12: Results from the use of stereotypes simulation experiment are depicted above. The bold red (darker gray) line indicates is a moving average for the no stereotyping condition. The bold yellow (light gray) line is a moving average for the stereotyping condition. Stereotyping requires fewer interacts to obtain an accurate partner model once the stereotypes have been constructed. Prior to stereotype construction, however, both methods perform the same. Note that the accuracy of the yellow (light gray) line does not decrease as much as the red line for later partners (P7-P19).129
- Figure 5.13: The graph depicts the accuracy of the classifier mapping a partner’s perceptual features to a stereotype model. As the robot interacts with additional partners the classifier has additional training data and its accuracy increases. The fact that the classifier accuracy goes to one indicates that the classifier correctly selects a stereotype model when given perceptual features. This does not mean that the model accurately reflects the partner.131
- Figure 5.14: The photos above depict the robot using stereotypes to select the correct partner model and then performed an action in a notional search and rescue environment. The first three photos depict the robot performing the action. The next two depict the targets and the robot’s view of the targets. When interacting with a person with the perceptual features of an EMT the robot retrieves the EMT stereotype model from memory. It uses this model to determine which of its actions the EMT would prefer and then does that action. The same is true for the firefighter.133
- Figure 6.1: This figure depicts two example outcome matrices for the cleanup of a toxic spill and the rescue of victims by a human and a robot. During any one interaction, both individuals choose to either rescue a victim or clean up a hazard. The outcomes resulting from each pair of choices are depicted in the cells of the matrix. The human’s outcomes are listed below the robot’s outcomes. In the leftmost matrix, the outcomes for the human and the robot are independent of the other’s action selection. In the rightmost matrix, the outcomes of the human and the robot largely depend on the other’s action selection.141
- Figure 6.2: Three dimensions of interdependence space are depicted above (Kelley et al., 2003). Interdependence theory represents social situations computationally as an outcome matrix within this interdependence space. The dimensions depicted above are interdependence, correspondence, and basis of control. Planes within this space denote the location of some well-known social situations, including the prisoner’s dilemma game, the trust game, and the hero game. A matrix’s location allows one to predict possible results of interaction within the situation...142

Figure 6.3: The procedure (Kelley & Thibaut, 1978) for deconstructing a social situation is presented above. This procedure is an analysis of variance of the outcome matrix that deconstructs the raw outcome matrix into three new matrices (the BAC, MPC, and MJC) representing different forms of control over the situation's outcomes. The outcome values for each of these three matrices are produced from the raw outcome matrix by iteratively 1) adding the noted cells, 2) dividing by the number of actions, and 3) subtracting the individual's mean outcome value. The variances of each matrix type are generated by calculating the outcome range for each choice of behavior and each individual. Because this example is of an independent situation, the MPC and MJC matrices do not vary.144

Figure 6.4: A mapping of interdependence space location to outcome matrix transformation.....151

Figure 6.5: This figure depicts the algorithmic process contributed by this work. The process consists of six steps. The first step generates an outcome matrix. The second step analyzes the matrix's variances. The third step computes the situation's interdependence space dimensions. These two steps constitute the process of situation analysis. The fourth step selects a transformation and in the fifth step, the transformation is applied to the outcome matrix resulting in the effective situation. Steps 4 and 5 constitute the transformation process. Finally, an action is selected.153

Figure 6.6: The simulation environment used for the cleanup and rescue experiment is depicted above. The experiment required that a teleoperated robot rescue victims while an autonomous robot performs a cleanup. Experimental conditions included independent versus dependent situations and the use of our situation analysis algorithm versus a control strategy. The teleoperation interface used by the human is depicted the right.....156

Figure 6.7: The procedures used to create and use outcome matrices are depicted above. The left side details the procedure used to generate Table 6.3.

This procedure first iterates through all matrices in each areas l_{hl}, l_{hh}, l_l and then iterates through the set of transformations to produce the matrix the robot will use to select actions. The middle procedure first creates a random number of victims and hazards. Next, an independent and dependent matrix is created from the number of victims and hazards. Finally, in the control conditions, max_own is used to select an action. In the test procedure, situation analysis is used to select an action. The right most procedure, first generates a random matrix and then transforms the matrix with respect to a control matrix or uses situation analysis. The robot selects an action from the transformed matrix. The interaction example at the bottom denotes the method used to determine how much outcome each individual receives from the presentation of an outcome matrix.158

Figure 6.8: Results for the cleanup and rescue experiment are presented above. The line graph portrays the net outcome for each condition. The bars depict the number of hazards and victims retrieved. Hazards cleaned are shown above the number of victims rescued. The left two bars and line points depict the independent conditions for both the test and the control robot. In these conditions both the control and test robot perform equally well. The right two bars and line points examine the dependent situation. Note that in this situation the test robot outperforms the control robot.160

Figure 6.9: Results of this second experiment are presented above. The second bar from the left indicates the net outcome when the situation analysis algorithm is used. The next four bars are the controls for the experiment. Error bars indicate 95% confidence interval. Analyzing the situation resulted in the greatest net outcome of when compared to the control strategies. The leftmost bar portrays the maximum possible net outcome. 164

Figure 7.1: Kelley and Thibaut noted that relationships can also be presented within the interdependence space (Kelley & Thibaut, 1978). This figure presents their original mapping of relationships within the interdependence space. Kelley and Thibaut recognized that relationships can be described in terms of interdependence and correspondence, two of the same dimensions that are used to describe social situation.171

- Figure 7.2: An example of a diagnostic situation. The robot and the human are presented with a given situation. The robot selects an action according to a `max_own` transformation and predicts the outcomes resulting for both itself and the human partner if the human selects according to a `max_other` relational disposition. In the resulting interaction depicted below, the human actually selects according to a `max_own` relational disposition. The situation is diagnostic because different outcomes for the robot result from different relational dispositions.176
- Figure 7.3: An example of a non-diagnostic situation is presented above. The situation is non-diagnostic because the outcome pair is the same regardless of the human's transformation type. The top row presents the interaction hypothesized by the robot and the middle row presents the resulting interaction. The key point here is that this given situation does not distinguish between the human's differing relational dispositions.180
- Figure 7.4: An example of an inverted situation. The situation is inverted because the robot's outcome in the resulting interaction (`min_other`) is greater than the robot's outcome in the hypothesized interaction (`max_own`).181
- Figure 7.5: The example above uses the given situation from Figure 7.2 and demonstrates use of the algorithm from Box 7.1. The given situation is transformed by the robot and the human to produce an effective situation and finally an action. The action pair results in an outcomes for both the robot and its partner. In the hypothesized interaction (top row) the outcome pair is predicted. In the resulting interaction, the outcome pair is the result of an interaction between the robot and the human. These pairs of outcomes as well as the robot's transformation type are used as input to the algorithm which characterizes the situation as diagnostic.184
- Figure 7.6: The graph above depicts the percentage of diagnostic situations as a function of matrix size. We hypothesized that matrices with fewer actions would result in a smaller percentage of diagnostic situations than matrices with more actions. The trend is true regardless of the type of comparison made.187
- Figure 7.7: The graph above depicts the accuracy of the partner's relational disposition as a function of partner transformation type variability. We hypothesized that as the partner's transformation variability increased the algorithm's accuracy would decrease. The results above support our hypothesis.194
- Figure 8.1: An example of the trust fall. The trust fall is a trust and team-building exercise in which one individual, the trustor, leans back prepared to fall to the ground. Another individual, the trustee, catches the first individual. The exercise builds trust because the trustor puts himself at risk expecting that the trustee will break her fall.199

Figure 8.2: The figure visually depicts the reasoning behind the development of the conditions for trust.200

Figure 8.3: The graphs depict the interdependence space mapping of random situations. The left hand side depicts only the situations meeting the conditions for trust (red). The right hand side depicts both those situations meeting the conditions for trust and those not meeting the conditions (blue). We hypothesized that the situations meeting the conditions for trust would form a subspace in the right hand side graph. As can be seen, the situations meeting the conditions for trust are interspersed with situations not meeting the conditions. Hence, our hypothesis is false; the situations meeting the conditions for trust do not form a subspace of the interdependence space.205

Figure 8.4: The figure depicts 2D graphs of situations meeting the conditions for trust (left hand side) and situations not meeting the conditions for trust (right hand side). Comparison of the graphs to the right with the graphs on the left indicates no difference. Hence, in none of the 2D graphs does the space of situations meeting the conditions for trust form a subspace of the interdependence space separate from those situations that do require trust.206

Figure 8.5: Graphical depiction of the increase of our proposed trust measure with respect to increasing loss and probability of untrusting action selection. Our trust measure is a unitless measure which is proportional to the amount of loss and the probability of selecting the untrusting action. The measure is useful for comparing situations that require trust.215

Figure 8.6: The top of the diagram shows the laboratory setup for the most trusted partner experiment. Left photo shows the base position which is located about 10 feet in front of two containers representing cell blocks. The center position shows the robot at an observation position in front of riot prisoners. The right photo depicts the robot observing the escapee prisoners. The two lower diagrams depict the actions the robot performs in the experiment. In the left diagram the robot first moves to a position within view of the operator and then moves to state “yes” or “no” with respect to its partner preference. In the right diagram, the robot moves to observe either the riot prisoners or the escapee prisoners.221

Figure 8.7: Robot movement for stating “yes” to the operator’s question regarding its partner preference. The robot moves its neck up and down to state yes.222

Figure 8.8: Robot movement for stating “no” to the operator’s question regarding its partner preference. The robot moves back and forth in a half circle to indicate no.223

Figure 8.9: Examples of the robot's observation actions from the two (left and center) prisoner observation points. The image to the right depicts an experimental trial conducted under limited lighting.....226

Figure 8.10: Results from the selecting the most trusted partner experiment. When the robot uses the algorithm from Box 8.3 to select the most trusted partner the average outcome was 10.57. The control condition, in contrast, in which the robot used a *max_own* strategy to select its action without consideration of the partner resulted in an average outcome of -7.24. The difference between these two conditions was statistically significant ($p < 0.03$).227

LIST OF ALGORITHMS

		Page
Box 5.1	The algorithm above creates an outcome matrix from the input partner and self models. The algorithm operates by successively filling in the elements of the matrix. The function x is a mapping from partner features to ID.	80
Box 5.2	Algorithm for using partner and self models to create outcome matrices. The algorithm successively updates the partner models achieving greater outcome matrix creation accuracy. The function x maps partner features to a partner ID, y maps situation features to the robot's self model, and z maps partner features to a partner model.	101
Box 5.3	Our algorithm for stereotype creation. The algorithm takes a new partner model as input. It then creates clusters of all of the stored models. The cluster centroids will serve as the robot's partner stereotypes. In the function learning phase, the robot learns a mapping from partner's features to the stereotypes. This mapping can now be used to retrieve a stereotype given the partner's perceptual features.	121
Box 5.4	The stereotype matching algorithm uses the partner's features to retrieve a stereotyped partner model.	121
Box 6.1	An algorithm for the analysis of a social situation.	143
Box 7.1	The algorithm above characterizes situations in terms diagnostic characteristics. The robot type is used to determine the comparator that will be used. Next the outcomes are used in conjunction with the information from Table 7.2 to determine the characterization.	183
Box 7.2	The algorithm above characterizes the partner's relational disposition. It takes as input a series of interactions and outputs the partner's transformation type. The algorithm operates by iterating through each type of relational disposition and several interactions, predicting the outcomes that would result from interaction with the partner type in line 1. After interacting the algorithm in Box 7.1 is used to characterize the situation. The characterization is used to either rule the type out or, possibly, conclude that the hypothesized is the true type.	189
Box 8.1	The algorithm above depicts a method for determining whether a social situation requires trust. The algorithm assumes that the first individual is the trustor, the second individual is the trustee, the action a_1^i is the trusting action, and the action a_2^i is not a trusting action.	202

Box 8.2	The algorithm above depicts a method for measuring the trusted required by a social situation.	213
Box 8.3	A method for selecting the most trusted partner among several potential partners is presented.	218

LIST OF SYMBOLS AND ABBREVIATIONS

θ	Transformation
O	Outcome Matrix
O_G	Given Situation
O_E	Effective Situation
o	Outcome
A	Action Set
$a_j \in A$	Action
$u(a_j^1, \dots, a_k^N) \rightarrow \Re$	Utility Function
m	Partner Model
$m.A$	Action Set within a Partner Model
$m.u$	Utility Function within a Partner Model
s	Stereotyped Partner Model
α	Interdependence Dimension
β	Correspondence Dimension
γ	Basis of Control Dimension
δ	Symmetry Dimension
$\langle \alpha, \beta, \gamma, \delta \rangle$	Interdependence Space Tuple
$O_E = f(O_G, \theta)$	Transformation Process
Y	Diagnostic Set
D	Diagnostic Situation
I	Inverted Situation

N	Nondiagnostic Situation
$R(a_1, a_2)$	Risk Function
$L(a_1, a_2)$	Loss Function
τ	Trust
T	Transformation Set
HRI	Human-Robot Interaction
AIBO	Artificial Intelligence roBOt
BAC	Bilateral Actor Control
MPC	Mutual Partner Control
MJC	Mutual Joint Control
I	Investment
R	Return
UT	Unreal Tournament
USARSim	Unified System for Automation and Robot Simulation
3D	Three Dimensions
2D	Two Dimensions
ID	IDentification
OpenCV	Open source Computer Vision
P0-P19	Partner's 0 through 19
EMT	Emergency Medical Technician

SUMMARY

Can a robot understand a human's social behavior? Moreover, how should a robot act in response to a human's behavior? If the goals of artificial intelligence are to understand, imitate, and interact with human level intelligence then researchers must also explore the social underpinnings of this intellect. Our endeavor is buttressed by work in biology, neuroscience, social psychology and sociology. Initially developed by Kelley and Thibaut, social psychology's interdependence theory serves as a conceptual skeleton for the study of social situations, a computational process of social deliberation, and relationships (Kelley & Thibaut, 1978). We extend and expand their original work to explore the challenge of interaction with an embodied, situated robot.

This dissertation investigates the use of outcome matrices as a means for computationally representing a robot's interactions. We develop algorithms that allow a robot to create these outcome matrices from perceptual information and then to use them to reason about the characteristics of their interactive partner. This work goes on to introduce algorithms that afford a means for reasoning about a robot's relationships and the trustworthiness of a robot's partners. Overall, this dissertation embodies a general, principled approach to human-robot interaction which results in a novel and scientifically meaningful approach to topics such as trust and relationships.

CHAPTER 1

INTRODUCTION

Many scientists have recently come to recognize the social aspects of intelligence (Byrne & Whiten, 1997; Sternberg, Wagner, Williams, & Horvath, 1995). In contrast to purely cognitive intelligence—which is most often described by problem-solving ability and/or declarative knowledge acquisition and usage—social intellect revolves around an individual’s ability to effectively understand and respond in social situations (Humphrey, 1976). Compelling neuroscientific and anthropological evidence is beginning to emerge supporting the existence of social intelligence (Bar-On, Tranel, Denburg, & Bechara, 2003; Bergman, Beehner, Cheney, & Seyfarth, 2003). Regardless of whether or not social intelligence is actually the dominant force behind intelligence, it is obvious that it is an important part of normal human development and intellect (Greenough, Black, & Wallace, 1987; Salzinger, Feldman, & Hammer, 1993). From the perspective of a roboticist it then becomes natural to ask how this form of intelligence could play a role in the development of an artificially intelligent being or robot. As an initial step one must first consider which concepts are most important to social intelligence.

One fundamental concept is the relationship (Gardner, 1983). In order to explore the possibility of developing a socially intelligent robot it will be necessary to computationally model and understand precisely what constitutes a relationship. A relationship is defined by the types and extent of influence one person has on another—their interdependence (see Appendix A for a complete glossary of terms) (Kelley & Thibaut, 1978). Relationships, moreover, are dynamic with each interaction among

individuals having the potential to alter the nature of the relationship (Rusbult & Van Lange, 2003). Similarly, the present state of the relationship will strongly guide the selection of behaviors while interacting. Kelley and Thibaut theorized that an individual will adjust its interactive behavior based on its perception of a pattern of outcomes, i.e. reward minus cost. They went on to develop interdependence theory, a conceptual skeleton for the study of relationships.

Due to the complexity of maintaining, judging, and updating multitudes of relationships over long periods of time, it becomes necessary to characterize the impending reliability of a relationship with respect to the social environment (Lewicki & Bunker, 1996; Luhmann, 1990). Trust serves this purpose. Trust enables an individual to gauge the risks associated with interacting with another agent (Kollock, 1994; Luhmann, 1979, 1990). Trust also allows an individual to estimate or predict the likelihood of future behaviors being employed (Gambetta, 1990). Finally, in the presence of trusted relations, an agent or robot's abilities may be augmented by the other specialties of the group, thus creating a collective that is more survivable than any single individual (Prietula, 2001).

If trusted relations are important, then the social situations that generate these relations are similarly vital. Later work by Kelley et al. outlined a number of canonical social situations and their key interpersonal properties (Kelley et al., 2003). For humans at least, interaction is often causally determined by the type of social situation in addition to each individual's personal responses to the situation (Rusbult & Van Lange, 2003). Their work also demonstrates that relationships develop from an accumulation of interaction in a variety of social situations (Kelley, 1979). This dissertation details a

general, computational framework for a robot or agent to represent and reason about both social situations and the relationships that develop from interaction with a partner.

1.1 Motivation

Human intellect has evolved and continues to evolve in a medium of social interaction. Moreover normal human brain development requires the nurture and support of social relations (Perry, 2001; Perry & Pollard, 1997). Clearly, if some of the goals of artificial intelligence are to understand, imitate, and interact with human-level intelligence then researchers must also explore the social underpinnings of this intellect. The goal of this work is to investigate the effect of characterizing the trustworthiness of social relationships on a robot's ability to understand, learn from, and interact within its social environment. There are many reasons why this endeavor is of value.

Social perception is important for robots operating in complex, dynamic social environments. For humans, social perception may include recognition of intent, attitude, and temperament and is a basic developmental skill in children (Pettit & Clawson, 1996). An individual's perception of his or her social environment is a crucial precursor to intelligent social action and interaction (Field & Walden, 1982; Travis, Sigman, & Ruskin, 2001). This perception allows the individual to judge the potential risks and rewards each of its relationships presents. In this work social perception focuses on characterizing a robot's relationships as worthy of trust. It is believed that by characterizing relationships in this manner a robot will have advantages in terms of learning and performing tasks in complex, dynamic environments. Moreover, in a suitable learning paradigm these advantages will be quantifiable as overall task performance. This work is motivated by the desire to develop robots with some

rudimentary understanding of social situations in the hope that they will then be better suited to operate in social environments. Learning within social environments is another critical aspect of social intelligence. Social environments offer developing animals the opportunity to learn from several different individuals in many different ways (Russen, 1997). This diversity of learning has an impact on the animal's ability to perform tasks critical to its survival. Current techniques in artificial intelligence tend to restrict a robot's source of learning to a single instructor and/or instruction signal (Mitchell, 1997; Sutton & Barto, 1998). The proposed research intends to consider instruction from multiple relations. It is believed that richer and more valuable guidance will be possible when learning from several relations, each with unique expertise. Moreover, social situations afford opportunities to learn not only about a specific task, but also about one's relationships, and which actions are best suited to build those relationships.

Finally, social behavior is vital for social intelligence. The challenge of creating robots that behave properly in a social environment is an important issue for robotics. As robots leave the lab and enter people's homes and families, it becomes critical that these artificial systems interact with humans in an appropriate manner. Because the actions of an embodied robot may entail risk for the robot's interactive partners, it is critical that the embodied robot consider the social costs of a potential action. The issues and problems associated with trust are of particular concern when a person expects to rely on a robot for their well-being. The research delineated in this dissertation is significant in that these issues will be examined in detail and for the first time in this context.

The proposed research focuses solely on relationships between a robot and a human. Multi-robot or relationships between simulated agents, although interesting, are not

addressed outside of the related work. Further, this dissertation proposes to investigate trusted relationships from the perspective of the robot, not the human. Hence, our intention is to ask questions such as can a robot be made to trust a human, rather than can a human be convinced to trust a robot. Although the latter is certainly of interest, a human-centric dissertation is not the primary motivation of the author.

We hope that the results of this work will be broadly applicable both within the artificial intelligence community and in other communities. Within artificial intelligence, it may be possible to describe many patterns of interaction between agents using interdependence theory. This broad applicability might extend to expert systems, planning, natural language understanding, and even perception. Moreover, an exploration of trust, relationships, and interaction from the perspective of artificial intelligence may have consequences for sociologists, social psychologists, and relationship researchers. The results and tools generated by this dissertation might serve these other disciplines in the same manner that other work in artificial intelligence has served their research (Axelrod, 1984; Bainbridge et al., 1994 for a review). While the experiments were performed in a human-robot interaction domain, we believe that the results are generalizable beyond human-robot interaction and robotics.

1.2 Principal Research Question

What effect will characterizing the trustworthiness of social relationships and of social situations have on a robot's ability to select actions?

As explained above, developing a framework that will allow a robot to characterize its social relationships may afford the robot advantages in its ability to select actions. This

dissertation determines to what extent a robot's characterization of its social environment aids in selecting actions.

From the principal research question the following subsidiary questions emerge. The discussion below each question describes why the solution to the subsidiary question is vital to solving the principal research question.

- 1) **What effect will the development of a theoretical framework that allows a robot to represent social situations and recognizing situations that require trust have on the robot's ability to select actions?**

The characterization of social relationships by a robot will require a computational method for representing and reasoning about the social situations that constitute the development of a relationship.

- 2) **What effect will deliberation with respect to the social situation have on the robot's ability to select actions?**

Social deliberation involves the consideration of one's social environment. As part of this subsidiary question, we develop a framework for social action selection that transforms the social situation perceived by the robot into a situation on which the robot will act. Moreover, we demonstrate that our framework for social action selection can include the robot's social dispositions.

- 3) **What effect will algorithms, developed as part of the theoretical framework of social situations, that allow a robot to represent its relationship with its human partner and to characterize these relationships in terms of the trust have on the robot's ability to select actions?**

For this subsidiary question, we introduce methods to develop models of the partner and to use these models. Social psychological research claims that relationships develop from a honing of one's model of their partner resulting from an accumulation of social interaction with the partner (Kelley et al., 2003; Rusbult

& Van Lange, 2003). This subsidiary question tests that claim and ties together many of the concepts presented throughout the dissertation.

1.3 Objectives

The principal objective of this research is to determine the role and impact of trust-characterized relationships on a robot's ability to perform tasks. Towards this goal, many novel and scientifically meaningful milestones will be accomplished. This dissertation will also provide insight into the phenomenon of trust and social relationships as well as a formal basis for developing a robotic implementation. Specifically, this research makes the following contributions:

- A general, computational framework implemented on a robot for representing and reasoning about social situations and interaction based on interdependence theory;
- A principled means for classifying social situations that demand trust on the part of a robot and for measuring the trust required by the situation in which the robot interacts with a human;
- A methodology for investigating human-robot interaction theory;
- A computational framework for social action selection implemented on a robot but generalizable beyond robotics. This framework will employ our computational representation of social situations in a manner suitable for a robot or simulated agent;
- An algorithm that allows a robot to analyze and characterize social situations;
- Methods for modeling the robot's human partner and characterizing a robot's relationship with the partner.

1.4 Dissertation Outline

The current chapter has introduced the focus of this dissertation as well as describing its motivations, contributions, and research questions. The next chapter surveys the relevant research and theory related to this research problem. Chapter 3 develops a methodology for investigating and developing HRI theory. Chapter 4 introduces our computational framework that allows a robot to represent and reason about its social interactions with a human partner. Chapter 5 presents algorithms and results that allow a robot to construct representations of its interactions and to learn from experience with a human partner. In chapter 6 we present an algorithm that allows a robot to characterize its social environment. In chapter 7 we detail algorithms and results of partner and relationship modeling. Chapter 8 explores trust, presenting definitions, computation methods for characterizing it and experimental results. Finally, chapter 9 offers conclusions.

CHAPTER 2

CHARACTERIZING HUMAN-ROBOT SOCIAL RELATIONS: A REVIEW

Recently scholars from a variety of fields have come to recognize the importance of the social environment in the development (Perry & Pollard, 1997), maintenance (Cross & Borgatti, 2000), evolution (Byrne & Whiten, 1997) and even the definition of intelligence itself (Gardner, 1983). Nicholas Humphrey, one of the earliest proponents of the importance of the social environment, observed that animals, most notably the higher primates, seem to possess abilities which exceed by far the necessities of their natural environments (Humphrey, 1976). He convincingly argued that social skills are the foundation of human intellect. Nevertheless, the process of relationship building that humans rely on from infancy has yet to be fully examined and adapted to relations between humans and robots. If it is true, as many anthropologists, psychologists, and sociologists claim, that the social environment plays a critical role in the existence of human intellect, then it becomes absolutely essential for roboticists to investigate and develop viable mechanisms by which a robot can manage a similar social environment. One important first step towards this goal is to examine the development of relationships by robots and the characterization of these relationships in terms of trust. This chapter reviews relevant work from many disciplines, highlighting gaps in the existing field of knowledge that this dissertation proposes to explore and detailing in depth the most pertinent literature.

2.1 Human-Robot Interaction

Human-robot interaction (HRI) is an emerging field of study that blends aspects of robotics, human factors, human computer interaction, and cognitive science (Rogers & Murphy, 2001). HRI is primarily concerned with the details of how and why humans and robots interact (see Fong, Nourbakhsh, & Dautenhahn, 2003 for a review). HRI touches on a wide variety of topics from the detection of human emotion (Picard, 2000) to human-oriented behavioral design (Arkin, Fujita, Takagi, & Hasegawa, 2003). This section begins by first reviewing the mechanics of human-robot interaction. Next, process models of interaction are explored. Finally, HRI methodology is reviewed.

2.1.1 Interactive communication

For humans, interaction is natural (Sears, Peplau, & Taylor, 1991). Robots, on the other hand, lack the basic competencies required to successfully interact with humans (Fong, Nourbakhsh, & Dautenhahn, 2003). Speech recognition and synthesis are a means of communication that can be used to facilitate human-robot interaction.

Speech synthesis has had a long and successful history within artificial intelligence (see Lemmetty, 1999 for a review). Recent work involving robots has focused on development of mechanical vocal cords for human mimetic speech generation by a robot (Shintaku et al., 2005). For other robots, such as the Sony QRIO, speech recognition is integrated with the robot's control architecture (Sony Corporation, 2006). Several software packages for speech recognition exist (Microsoft Speech SDK 5.1 information page, 2006; Open Mind Speech, 2006). Moreover, improvements in recognition have allowed many commercial applications for recognition technologies to flourish (Karat,

Vergo, & Nahamoo, 2007). Speech synthesis applications, like speech recognition, has become an established technology (Robert, Clark, & King, 2004). In addition to the raw perceptual challenges of recognizing and producing communicative acts for a human, an HRI researcher must also consider the style of interaction.

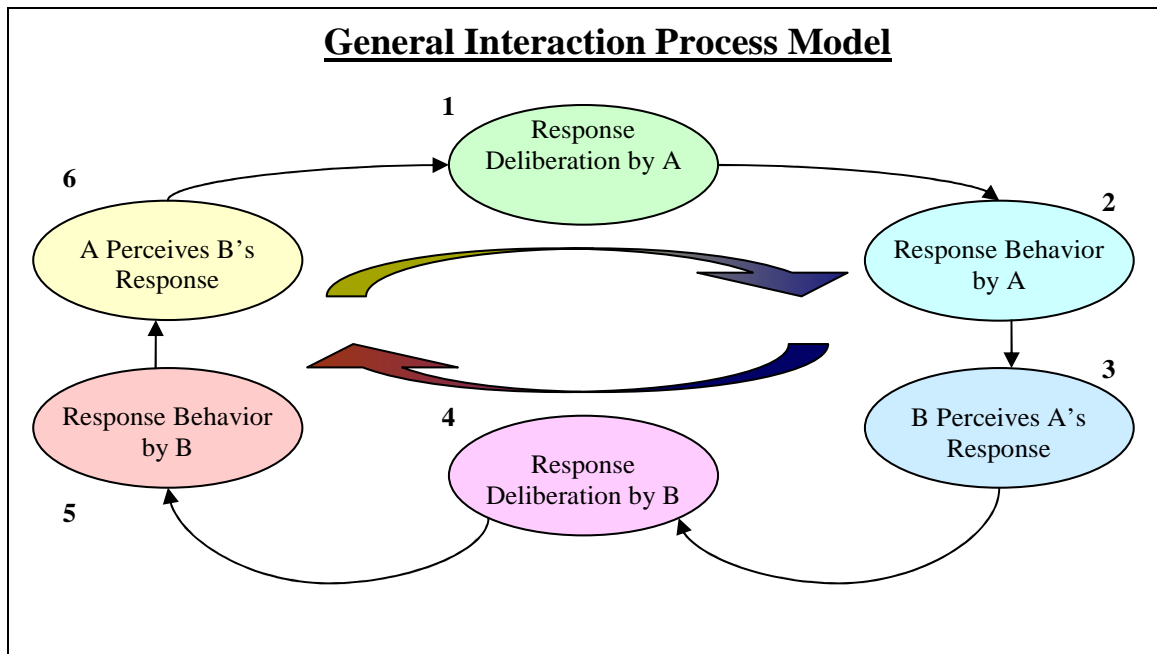


Figure 2.1 A process model for turn-taking interaction is depicted above. This style of interaction involves iterative distinct responses.

2.1.2 Styles of interaction

Dyadic social interaction (see Appendix A for glossary of terms) typically involves either a concurrent style of interaction or a turn-taking style of interaction (Kelley, 1984). A concurrent style of interaction requires that the individuals select their actions at the same time. Turn-taking style, on the other hand, allows each individual the opportunity to observe the social action of their partner before selecting their own social action. The major difference between these styles is thus the timing of interaction between members of the dyad, and not what is expressed. Either style of interaction could therefore be used

to investigate the development and characterization of human-robot relationships. We use both the turn-taking and concurrent style of interaction for experimentation.

As will be discussed in section 2.2.1, trust (Kollock, 1994) and relationships (Kelley et al., 2003) in general develop from repeated interaction among individuals. Hence, a process model for turn-taking can be used to describe the cycle by which a human and a robot interact repeatedly over some number of iterations. Figure 2.1 depicts a cycle from this general turn-taking process model. This model relates to the transition network style of dialogue models (see Green, 1986 for a review). In Green's model vertices represent the states of the dialogue between the user and the robot. Edges determine the transition from one dialogue state to another in this model.

The general process model from Figure 2.1 describes an iterative procedure. Assume that *A* is the human, *B* is the robot, and that the interaction begins arbitrarily with (1) deliberation by *A* concerning which interactive behavior to employ. First, (2) *A* determines which interactive behavior to employ and uses the behavior. Next (3) *B* perceives *A*'s interactive behavior. Then (4) *B* deliberates to determine the proper response based on knowledge of *A*'s interactive behavior. Next (5) *B* produces an interactive behavior and finally (6) *A* perceives *B*'s interactive behavior continuing the cycle. In this model the robot selects interactive behaviors at a higher, deliberative level in the robot architecture.

2.1.3 HRI methodology

Fong et al notes that two design methodologies dominate HRI research: the functionally designed approach and the biologically inspired approach (Fong, Nourbakhsh, & Dautenhahn, 2003). Functionally designed methodologies focus on social task

performance without any correlation to living creatures. Biologically inspired methodologies tend to mimic or simulate their biological counterparts. This dissertation will employ a biologically-inspired approach, drawing heavily from work in psychology and sociology.

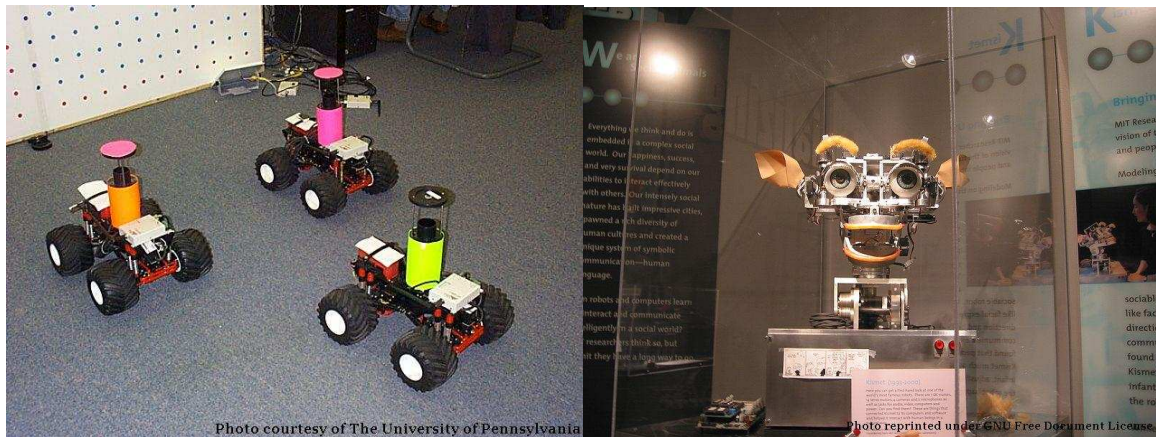


Figure 2.2 Although extremely mobile, robots such as the Clodbusters (Hsieh et al., 2007) on the left have been designed with little capacity for interaction. Kismet to the right, on the other hand, has been designed specifically for interaction. Although many degrees of freedom control its facial expressions the robot was not designed for general purpose motion over long distances.

Methodological evaluation of an interactive robot is often influenced by the capabilities of the robot. Mobile robots have traditionally tended to possess great capabilities for exploring their environment, yet little capacity for interaction with humans (Kortenkamp et al., 1998; Thrun et al., 1998) (Figure 2.2 gives an example). On the other hand, Kismet (Figure 2.2 left)—an actuated expressive robotic head—is capable of an impressive array of interaction yet largely stationary (Breazeal, 2002).

Field testing is another method of evaluation often employed (Pineau, Montemerlo, Pollack, Roy, & Thrun, 2003). Interactive robots in the field range widely with regard to purpose and capabilities (e.g. Nourbakhsh, 1998; Thrun, Schulte, & Rosenberg, 2000). Proof of concept systems are common (Fong, Nourbakhsh, & Dautenhahn, 2003). One long-term goal of this work is to develop robots capable of building trusted relationships

with people in need of assistance. Several HRI research projects have explored the use of robots as assistive navigators for humans. One of the earliest was Horswill's Polly robot which gave tours of an office environment (Horswill, 1998). Nourbakhsh also developed a robot tour guide for an office environment (Nourbakhsh, 1998). Thurn et al. investigated the use a robot as a museum tour guide (Thrun, Schulte, & Rosenberg, 2000). Stoychev and Arkin explored using robots for office delivery tasks (Stoychev & Arkin, 2001). Researchers have recently begun to study the prospect of using robots to assist the visually impaired (Lacey & Dawson-Howe, 1998; Shoval, Ulrich, & Borenstein, 2000). Autistic children have been one area of focus (Scassellati, 2000; Werry & Dautenhahn, 1999). Others have investigated using robots to assist the elderly (Pineau, Montemerlo, Pollack, Roy, & Thrun, 2003) and the disabled (Mataric', Eriksson, Feil-Seifer, & Winstein, 2007).

In many ways interaction describes the surface of relationship building. Further, the details of interpersonal interaction often causally relate to the social situation that spawned the interaction (Kelley et al., 2003). Metaphorically, the social situation serves as interactive scaffolding from which an interpersonal relationship develops. The next section will therefore review research from psychology and sociology created to explore the nature of interpersonal relations.

2.2 Interpersonal Relations

For humans, social relations form the environmental fabric of our existence (Byrne & Whiten, 1997; Gardner, 1983; Humphrey, 1976; Travis, Sigman, & Ruskin, 2001). This research investigates social relationships from the perspective of a robot. A necessary starting point for this investigation is a consideration of which, if any, theories of human

relationships are relevant to this endeavor. This section reviews theories pertaining to the development, maintenance, and continuation of human interpersonal relations.

2.2.1 Relationship theory

Social interaction is defined as influence—verbal, physical, or emotional—by one person on another (Sears, Peplau, & Taylor, 1991). Relationships develop from interaction between two individuals or dyads. Several theories describing why and how relationships develop have been proposed. An investigation of relationships between humans and robots requires an underlying conceptual framework to support the design methodology. Because this is a biologically inspired approach we turn to related work from social psychology for this theoretical framework. Note that the purpose of this theoretical framework is two-fold: first, it is necessary to have a basis for understanding the actions a human will choose; second, it is necessary to have a basis for determining the correct robot responses to a given social situation. The theoretical framework selected must provide both. This section first reviews competing alternatives from social psychology, selects one, and then details the theory from the psychological perspective describing why it is the correct choice for implementation on a robot.

Social penetration theory views the development of a relationship as a process of increasing self-disclosure (Altman & Taylor, 1973). As a relationship develops the individuals in the relationship confide, share, and offer more personal information to the other person. Supporters of the theory claim that social penetration theory successfully explains several aspects of relationships such as each individual's dependence on the other individual in close relationships (Sears, Peplau, & Taylor, 1991). Critics of penetration theory claim that the theory is not supported by data, fails to explain high levels of

reciprocity and altruism in middling relationships, and does not account for differences in gender, culture, or race (Griffin, 1997). As a model of relationships for robots, penetration theory assumes the presence of vast perceptual and behavioral competencies that would be difficult to develop. For example, recognizing disclosure would likely require affect recognition, topical understanding, contextual perception and understanding, and possibly much more. Moreover, the high level descriptions of social penetration theory developed for psychologists would be difficult to interpret and implement on a computer.

Uncertainty reduction theory focuses on the effect and usage of communication among humans in relationships as a means for reducing uncertainty in our social environment (Berger, 1987). An axiomatic approach, uncertainty reduction theory delineates the connection between uncertainty reduction and social psychological traits such as reciprocity and similarity. Supporters of uncertainty theory claim that the theory explains aspects of communication within human relationships not well addressed by other theories. Criticisms of uncertainty reduction theory often focus on its axioms, typically claiming one or several of them are invalid (Griffin, 1997). As a theory of interpersonal relations for robots, uncertainty reduction theory could potentially be used as a means for modeling newly developed relationships. This theory, however, is not adequate (nor was it meant to be) for modeling close relationships.

Interdependence theory, as will be shown, is adequate for modeling both superficial and intimate interpersonal relationships. It also models relationships computationally in a manner suitable for implementation on a robot. Interdependence theory is a social psychological theory developed by Kelley and Thibaut as a means for understanding and

analyzing interpersonal situations and interaction (Kelley & Thibaut, 1978). It began as a method for investigating group interaction processes and evolved over the authors' lifetimes into a taxonomy of social situations categorizing interpersonal interactions (Kelley et al., 2003; Kelley & Thibaut, 1978). Moreover, interdependence theory is considered by some to be the most influential social psychological theory for this purpose (Sears, Peplau, & Taylor, 1991) and will thus form the theoretical framework for this dissertation. The term interdependence describes the effects interacting individuals have on one another. Interdependence theory is based on the claim that four variables dominate interaction: 1) reward, 2) cost, 3) outcome, and 4) comparison level. Reward refers to anything that is gained in an interaction. Cost, on the other hand, refers to the negative facets of the interaction. Outcome describes the value of the reward minus the cost. An individual will adjust its behavior based on its perception of a pattern of outcomes. Comparison level describes an individual's tendency to compare the actual outcomes from a relationship with the individual's expected outcomes. Two of interdependence theories four core variables (reward and cost) relate to terms long familiar to artificial intelligence researchers (Sutton & Barto, 1998).

Critics of interdependence theory often state that 1) it ignores the non-economic aspects of interpersonal interaction such as altruism and 2) that it assumes people are rational, outcome maximizers. Kelley responds to these criticisms directly, stating that the noneconomic aspects of interaction can also be included in a description of a person's outcomes and that the theory does not presume either rationality or outcome maximization (Kelley, 1979). Rather, as will be explained shortly, individuals often

transform social situations to include the irrational aspects of socialization such as emotion or social bias.

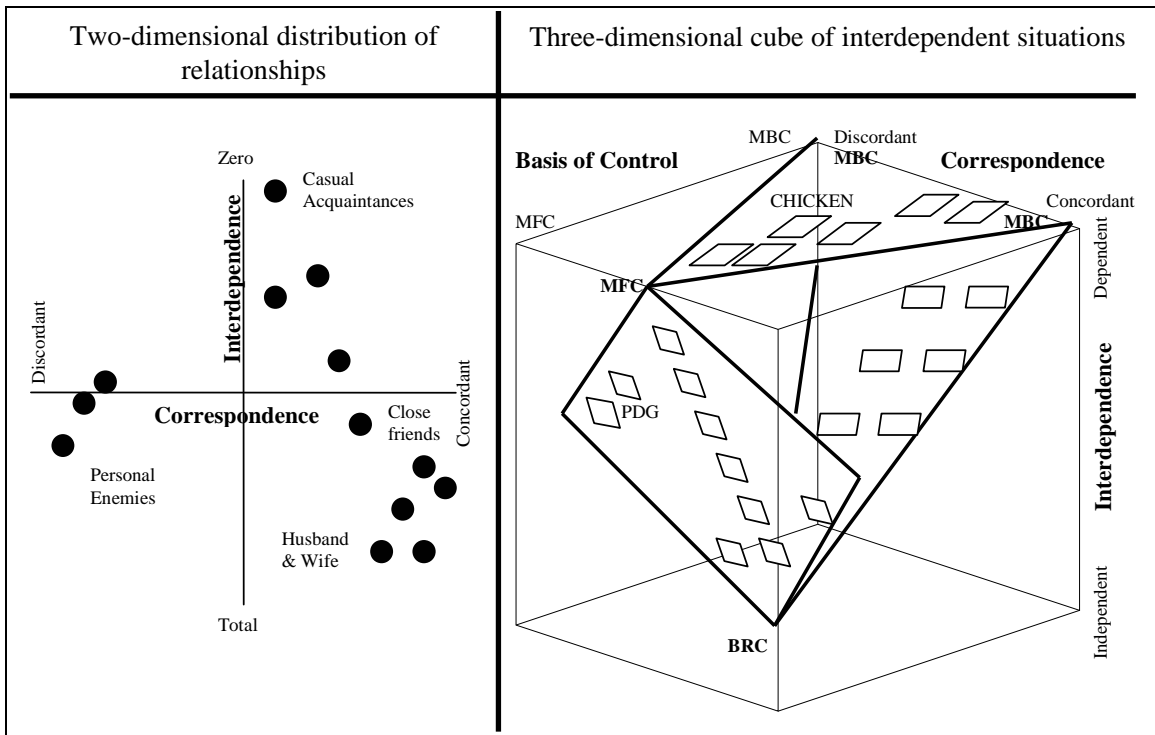


Figure 2.3 The distribution to the left depicts a two dimensional cross-section of the interdependence space. Here some prototypical relationships are described in terms of their interdependence and correspondence. To the right, a three dimensional distribution adds the basis of control dimension. This cube depicts prototypical interpersonal situations such as the prisoner's dilemma game (PDG) and trust game as squares within the cube. These prototypical situations often occur on planes within the cube (adapted from Kelley, 1979 and Kelley et al., 2003).

Interdependence theory serves as the conceptual skeleton for analyzing *interactive* or *social situations*. A social situation describes the social context surrounding an interaction between individuals (Rusbult & Van Lange, 2003). Recently social psychologists have developed an atlas of canonical social situations (Kelley et al., 2003). Figure 2.3 depicts a variety of social situations in a space termed the interdependence space. All of the social situations described within interdependence theory are discrete events that map to a location within interdependence space. The prisoner's dilemma

game and the trust game are examples of social situations that can be mapped to a portion of the interdependence space.

The interdependence space is a four dimensional space (Figure 2.3 right only depicts three of the four dimensions) that describes all social situations. These dimensions are interdependence, symmetry, correspondence, and basis of control. The interdependence dimension describes the extent that each partner's outcomes are influenced by the other partner's actions. The symmetry dimension describes the degree to which the partners are equally dependent on one another. The correspondence dimension describes the extent to which each partner's outcomes are consistent with the others. The basis of control dimension describes the ways in which each partner affects the other's outcomes. Table 2.1 lists each dimension and describes the maximal and minimal values for each dimension. Dimensional values for situations within the space are derived from reward, cost and outcome values for each possible action in the social situation. These values are typically depicted in an outcome matrix such as the one in Figure 2.4. Outcome matrices are not typical linear algebraic matrices and are equivalent to the normal form game representation (Chadwick-Jones, 1976). This example matrix describes a social situation involving two individuals labeled one and two. In this example, both individuals interact by selecting one of two behaviors: a_1^1 and a_2^1 for individual one and a_1^2 and a_2^2 for individual two. Cells $_{xy}o^1$ thru $_{xy}o^2$ denote the outcome values for each combination of behaviors selected. Thus, $_{21}o^1$ describes the outcome value for individual one if individual one selects action a_1^1 and individual two selects action a_2^2 . Likewise, $_{21}o^2$ describes the outcome value for individual two resulting from the same action selection.

Table 2.1 The dimensions of interdependence space are listed with descriptions of the maximal and minimal values and examples at the extremes.

Interdependence Space Dimensions		
Dimension	Range	Description at extremes
Degree of interdependence	Complete interdependence—Zero interdependence	Outcomes entirely depend on the actions of the partner; Outcomes are independent of the partner's actions
Symmetry	Symmetric dependence—Unilateral dependence	Both partners can equally effect the other; One partner has greater control over the outcomes of the other partner
Correspondence	Corresponding interests—Conflicting interests	Partners act for each other's mutual interest; Partners act in the opposite of the partner's interest.
Basis of control	Outcome exchange—Outcome coordination	Partners receive favorable outcome by exchanging one action for another; Partners receive favorable outcome by coordinating joint actions

An Outcome Matrix

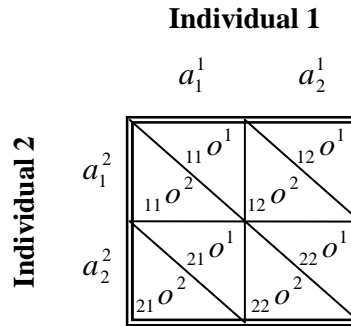


Figure 2.4 An example outcome matrix is depicted. The term $_{xy}o^1$ denotes the first individual's outcomes and the terms $_{xy}o^2$ denote the second individual's outcomes. Outcomes result from the selection of an action pair by each individual.

The left hand side of Figure 2.3 also shows the location within interdependence space of some typical relationships. One of interdependence theory's core premises is that a relationship develops from a culmination of interaction and a dyad's movement and decisions with respect to the social situations faced (Kelley, 1984; Kelley et al., 2003; Kelley & Thibaut, 1978). This is a very important point which is worth restating. Interdependence theory claims that human relationships accrete from continued interaction between two people. For humans and robots, this dissertation examines whether or not a robot can model and predict the behavior of its human partner, allowing

it to characterize its relationship and alter its behavior accordingly. This dissertation does not explore the human psychology of relationship building with a robot. We leave that portion of this work for the future. It should also be noted that by controlling the social situation in which a human and a robot are immersed, we can guide the development of the relationship and the corresponding characterization of later relationships.

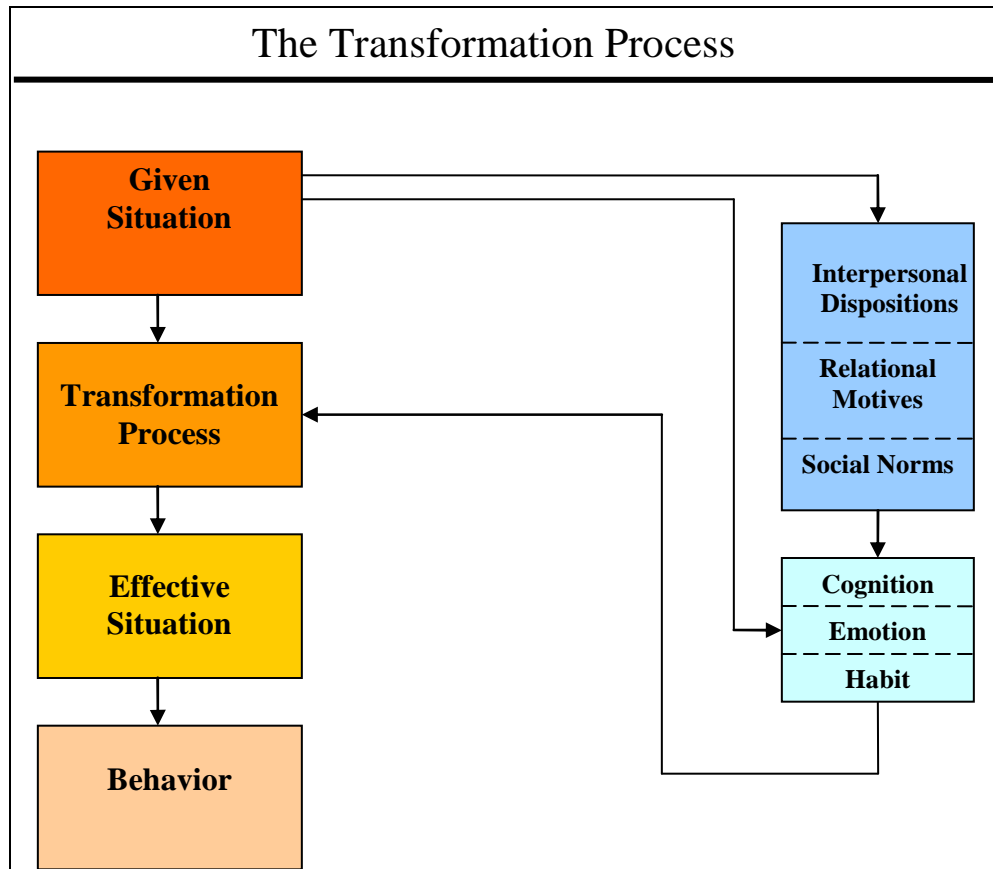


Figure 2.5 The transformation process (adapted from Rusbult & Van Lange, 2003). This process model transforms a given or perceived situation into behavioral selection by a person. The given situation is influenced by cognitive factors such as relational motives and emotions. These factors transform the perceived situation into an effective situation that the individual uses for selecting social behavior.

Given that a human and a robot are immersed in a social situation, interdependence theory describes a process by which social perception is transformed into social behavior. Reminiscent of the sense-plan-act paradigm (Bonasso, Kortenkamp, & Murphy, 1998), Kelley and Thibaut developed a transformation process (Figure 2.5) in which sensory

perception is transformed into action after being influenced by the agent's internal cognitive processes (Kelley & Thibaut, 1978). In this case, however, social situations are used as the quanta of perceptual input. In the vernacular of interdependence theory, the perceived situation is termed the given situation. The given situation is a perceived instance of one type of social situation. The given situation is perceived by the individual and then cognitively transformed, creating an effective situation on which action is based. Hence, the final product of this process is the effective situation. The effective situation represents outcomes that include many various aspects of the individual's own internal predilections. Behaviors are directly selected from the resulting effective situation. This process is illustrated in Figure 2.5. Several factors influence how the transformation process actually converts outcomes from the given situation to the effective situation. Examples include the individual's dispositions, motivations, and relational or social norms (Holmes & Rempel, 1989; Kelley, 1984; Kelley & Thibaut, 1978). Interpersonal dispositions are actor-specific response inclinations to particular situations across numerous partners (Rusbult & Van Lange, 2003). Motives, on the other hand, are partner- and situation-specific response inclinations. Social norms are rule-like, "socially transmitted inclinations" governing the response to a particular situation in some specified manner (Knight, 2001; Rusbult & Van Lange, 2003). Rusbult and Van Lange also describe a socially reactive mechanism by which children select behaviors based on the given situation without consideration of deliberative level social norms and motives of the transformation process (Rusbult & Van Lange, 2003). The computational methods described in this dissertation could therefore include connections to developmental robotics.

In addition to explaining how relationships evolve, interdependence theory can be used to describe the forces that govern whom an individual selects to engage in a relationship (Kelley & Thibaut, 1978). Simply put, interdependence theory posits that people are attracted to those that present the largest interactive outcomes. Researchers also describe attraction as a function of familiarity, competence, and proximity (Duck, 1973). Relationships may be motivated by many different reasons including kinship, sex, or survival (Wright, 1999). Cooperative relationships describe connections between individuals that offer the possibility of advantage for individuals which would not be present without the relationship (Knight, 2001; Trivers, 1971). It is believed that interdependence theory, as described above, can address the challenge of human-robot social behavior.

2.2.2 Relationship and social situation analysis

In addition to providing a means for modeling relationships computationally, interdependence theory also provides the computational methods necessary for analyzing relationships and social situations. Relationship and situation analysis is critical for a robot operating in dynamic and/or complex **social** environments. A robot must recognize the social impacts of each of its behavioral options, in addition to the impact on its own outcomes.

Situation analysis begins with the given social situation described in the preceding section. The given situation contains raw or unanalyzed outcomes for the social situation it represents. Situation analysis breaks down the given situation into three constituent components: the Bilateral Actor Control matrix (BAC), the Mutual Partner Control (MPC) matrix, and the Mutual Joint Control (MJC) matrix. The BAC matrix depicts the

extent to which each individual controls his or her own outcome. The MPC matrix, on the other hand, describes the manner in which the partner controls each individual's outcomes. Finally, the MJC matrix shows how each partner's outcomes are affected by a combination of their action and their partner's action. The result of this analysis is a description of the relationship or situation in interdependence space. Analysis also quantifies the extent to which each of the robot's behaviors influences the robot, the partner, and the robot and partner jointly.

Analyzing a Social Situation with Example

PROCEDURE:

- 1) Add cells 2) Divide by two 3) Subtract mean 4) Place result in the designated matrix cell

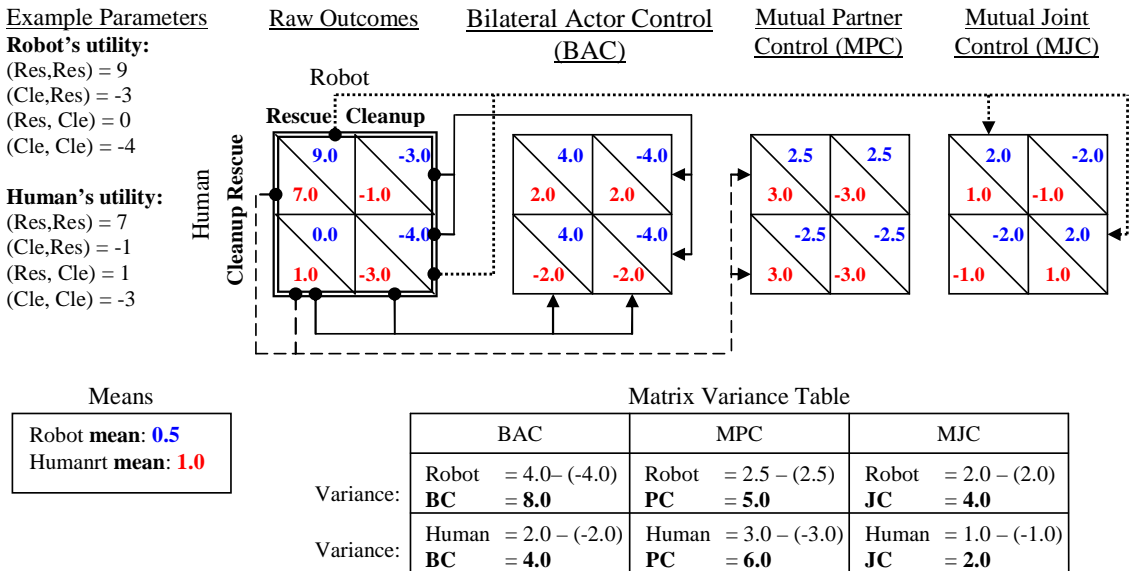


Figure 2.6 Example outcome matrices are depicted above. The original outcome matrix is on far left. These outcomes are converted into the BAC, MPC, and MJC by following the procedure listed at the top of the figure. The numbers for the raw outcomes are provided in the example parameters listed to the left. In this example the robot selects between rescue and cleanup actions. The procedure results in the matrix variance table listed towards the bottom of the figure.

The process of relationship and social situation analysis is described with an example and illustrated in Figure 2.6. If we consider the search and rescue domain, then both the human and the robot could select either an action to rescue a victim or to cleanup a

hazard. In this case, the actual numerical value for each outcome could be a function of the number of victims and hazards in the environment. Often the actual values within the cells of a matrix are less important than the relation of one cell to another cell. For example, it is typically more valuable to know which action in an outcome matrix provides maximal reward than it is to know the actual value of the reward. The first matrix depicts the raw outcome scores for each individual. Thus in this example, the robot rates its outcome for mutual rescue as 9 and the human rates its outcome for mutual rescue as 7 (the source of these ratings is discussed in the next chapter). Units represent some measure of satisfaction. From this matrix one can derive the Bilateral Actor Control (BAC) matrix, the Mutual Partner Control (MPC) matrix, and the Mutual Joint Control (MJC) matrix by adding the appropriate cells from the raw outcome matrix, dividing by two, and then subtracting the mean. The variance for each matrix is displayed at the bottom of the figure. Variance is calculated from the results in the matrices by measuring the difference in outcome from one behavioral option to the other. Once an analysis has been conducted a robot will have the necessary information to fully reason over the social impacts of the situation or relationship to determine the appropriate social action. Summed over many situations, these values would then describe the present state of the relationship between the robot and the human.

2.2.3 Social learning

Social learning encompasses many different types of learning, only some of which have been traditionally investigated by AI researchers. Our working definition for social learning will be *improvement with respect to some performance measure on some class of tasks with experience derived from a **social** environment*. A social environment is

defined as any environment with more than one social robots and/or humans (see Appendix A for glossary of terms). Social learning has come to include teaching (e.g. Angluin & Krikis, 2003; Jackson & Tomkins, 1992), learning by imitation (e.g. Billard, Epars, Calinon, Schaal, & Cheng, 2004; Schaal, 1999), learning by observation and practice (e.g. Bentivegna, Atkeson, & Cheng, 2004; Lin, 1992), learning about the social environment (e.g. Banerjee, Mukherjee, & Sen, 2000; Schillo & Funk, 1999; Schillo, Funk, & Rovatsos, 2000) and learning about one's role in the social environment (e.g. Crandall & Goodrich, 2004). Similarly, social reinforcement learning combines a traditional reinforcement learning paradigm, which is defined by the types of problems solved (Sutton & Barto, 1998), with traditional social learning (Isbell, Shelton, Kearns, Singh, & Stone, 2001). Abeel and Ng explored the challenge of developing an agent that could learning another agent's reward function via observation (Abbeel & Ng, 2004).

Rather than focusing on the computational aspects of learning, social psychologists have mainly focused on the social aspects of learning (Bandura, 1962; Sears, Peplau, & Taylor, 1991). Association learning is a general term for describing the learning of an association connecting the occurrence of one artifact to the occurrence of another and is well suited for the purpose of this research. Associations can be generated mapping the robot's specific behavioral choice to the perceived state of the partner and their relationship. Associations are used as a general mechanism for modeling and predicting a partner's interactive responses.

Credit assignment is another relevant concern when pursuing learning in a machine. Credit assignment is defined as "the problem of assigning credit or blame to the individual decisions that led to some overall result" (Cohen & Feigenbaum, 1982).

Diagnostic situations, attribution-directed activity, and clarification are used in interpersonal relationships to mitigate the challenges of the credit assignment (Holmes & Rempel, 1989; Rusbult & Van Lange, 2003). A diagnostic situation is a means by which an individual tests a credit assignment hypothesis to determine the proper assignment. Attribution-directed activity, on the other hand, allows an individual to attribute credit, temporarily, to some cause and then update this assignment later as additional evidence becomes available. Clarification simply motivates the individual to locate the credit-deserving aspect of the environment and typically occurs in fledgling relationships.

Stereotype learning is another important type of learning found in humans (Sears, Peplau, & Taylor, 1991). Stereotypes can be described as a manifestation of an interpersonal schema relating perceptual features to distinctive clusters of traits (Sears, Peplau, & Taylor, 1991). Stereotypes offer distinct computational advantages in terms of processing time for new stimuli and reaction time for previously encountered stimuli (Rusbult & Van Lange, 2003). Subgrouping is a stereotyping process by which information is organized into multiple clusters of individuals who are similar to one another in some way and different from others (Richards & Hewstone, 2001). Stereotyping is used in this research as a means for developing internal models for unknown partners. Stereotypes are used to bootstrap the process of partner model building allowing a robot make an educated guess about a new partner based on interactions with prior partners.

2.2.4 Connections to game theory

Section 2.2.1 briefly mentioned the equivalence of the normal form game representation and the situation outcome matrix. This section will discuss the similarities and differences of interdependence theory and game theory.

Game theory “is a bag of analytical tools” to aid one’s understanding of strategic interaction (Osborne & Rubinstein, 1994). In game theory, an equilibrium is a set of strategies in which no individual can unilaterally improve the outcome they receive. Interdependence theory, on the other hand, focuses on the development of relationships and the social situations from which these relationships grow. Game theory and interdependence theory both use the outcome matrix to represent social situations and interaction (Chadwick-Jones, 1976). In game theory these situations are limited by several other assumptions, namely: both individuals are assumed to be outcome maximizing; to have complete knowledge of the game including the numbers and types of individuals and each individual’s payoffs; and each individual’s payoffs are assumed to be fixed throughout the game. Interdependence theory does not make those assumptions. Because it assumes that individuals are outcome maximizing, game theory can be used to determine which actions are optimal and will result in an equilibrium of outcome. An equilibrium in game theory is an action or series of actions which do not depend on the actions of the opponent. Interdependence theory, because it makes no assumptions about how an individual will transform the given situation (i.e., maximize their own outcome, maximize their partner’s outcome, etc.) does not lend itself to analysis by equilibrium of outcomes. For this reason, situations are analyzed in terms of variance of outcome with respect to the actions selected by the dyad, as described in

section 2.2.2. This is a crucial difference between this dissertation work and previous investigations by other researchers using game theory to control the social behavior of an agent (e.g. Crandall & Goodrich, 2004).

This section has described many of the social psychological and hence biologically inspired underpinnings that serve as the theoretical basis for this research. In the next section the background and role of trust is reviewed in detail.

2.3 Using Trust to Characterize Relationships

Trust has been studied by a variety of researchers from several different fields (Rousseau, Sitkin, Burt, & Camerer, 1998). This section reviews the vast, variegated literature concerning trust. Our inquiry begins with the different definitions of trust. Next, the use of social situations in the evaluation of trust is examined. This section concludes by describing alternative methods for evaluating trust.

2.3.1 Definitions of trust

Early trust research focused on definitions and characterizations of the phenomenon. Morton Deutsch is widely recognized as one of the first researchers to study trust (Marsh, 1994). Deutsch, a psychologist, describes trust as a facet of human personality (Deutsch, 1962). He claims that trust is the result of a choice among behaviors in a specific situation. Deutsch's definition of trust focused on the individual's perception of the situation and the cost/benefit analysis that resulted. Hence, his definition bears close ties to interdependence theory. He also proposes the existence of different types of trust. Each type of trust is classified according to the situation in which it occurs. Trust as social conformity, for example, results from societal expectations of trust. Other types include

trust as despair, innocence, impulsiveness, virtue, masochism, faith, risk-taking, and confidence (Deutsch, 1973; see Marsh, 1994 for an overview).

Niklas Luhmann, another early trust researcher, provides a sociological perspective (Luhmann, 1979). Luhmann defines trust as a means for reducing the social complexity and risk of daily life. He argues that the complexity of the natural world is far too great for an individual to manage the many decisions it must make in order to survive. For Luhmann, trust is one method for reducing societal complexity. Because a trusting society has greater capacity for managing complexity, it can afford to be more flexible in terms of actions and experience. In addition to managing complexity, he claims that trust is a method for handling risk. Lewis and Weigert extend Luhmann's conceptualization of trust, adding emotional and cognitive dimensions (Lewis & Weigert, 1985).

Bernard Barber, another sociologist, defines trust as an expectation or mental attitude an agent maintains regarding its social environment (Barber, 1983; see Marsh, 1994 for an overview). He claims that trust results from learning in a social system and is used by an individual to manage its expectations regarding its relationships and social environment. Hence, trust is an aspect of all social relationships and is used as a means of prediction for the individual. Here again trust is defined in terms of social relationships open to exploration via interdependence theory.

Gambetta describes trust as a probability (Gambetta, 1990). Specifically, he claims that, "trust is a particular level of subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action and in a context in which it affects his own action" (Gambetta, 1990) pg216). Gambetta defines trust as a probabilistic assessment of another agent's intent to

perform an action on which the agent will rely. Because of its simplicity, this definition has not been without controversy (Castelfranchi & Falcone, 2000).

Rousseau et al. have examined the definitional differences of trust from a variety of sources (Rousseau, Sitkin, Burt, & Camerer, 1998) and concluded that trust researchers generally agree on the conditions necessary for trust, namely risk and interdependence. The work of Deutsch, Luhmann, Barber, and Gambetta has served as a starting point for many later investigations of trust.

Lee and See consider trust from the perspective of machine automation, providing an extremely insightful and thorough review of the trust literature (Lee & See, 2004). They review many definitions of trust and propose a definition that is a compilation of the many previous definitions. Namely, trust is *the attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability*. We use the definition for trust presented by Lee and See to generate a more conceptually precise definition of trust. We define trust in terms of two individuals—a trustor and a trustee. The trustor is defined as the individual doing the trusting. The trustee represents the individual in which trust is placed.

Trust is a belief, held by the trustor, that the trustee will act in a manner that mitigates the trustor's risk in a situation in which the trustor has put its outcomes at risk.

The preceding will be the working definition for trust used for the proposed dissertation. Methods for quantifying trust are discussed in the following section.

2.3.2 Using social situations to evaluate trust

As detailed in the next section, many different researchers have generated many different computational models of trust. Our approach is to show that if a general, principled framework for social situations is used in conjunction with a well-defined definition for trust, then we are able to segregate those situations that require trust from situations that do not require trust naturally and without modification of our framework. In other words, as will be shown in chapter 8, our framework for situation-based interaction implicitly contains mechanisms for determining if and how much trust is necessary for a given social situation. We show that given the ability to recognize and gauge the trust required by a social situation, a robot can then use this information to characterize a relationship with a particular individual or a particular type of situation. Trust will be measured in terms of risk calculated using a loss function (Risk, 2007). In this manner, we demonstrate that many of the models proposed by other researchers are actually special cases of our method.

Before examining alternative approaches for evaluating trust, we will first detail methods that focused on the social situation itself. These methods are widespread both within neuroscience (Quervain et al., 2004; Sanfey, Rilling, Aronson, Nystrom, & Cohen, 2003) and experimental economics (Berg, Dickhaut, & McCabe, 1995; McCabe, Houser, Ryan, Smith, & Trouard, 2001) and typically involve social situations “instantiated” in a variety of real world experiments with the aim of exploring the phenomena of trust. The method employed by King-Casas et al. used a situation in which two human players iteratively interact for ten rounds exchanging money as an investor and as a trustee. In each round the investor selects some proportion of money to invest (I) with the trustee.

The money appreciates ($3I = R$). Finally the trustee repays a proportion of the total amount (R) back to the investor. King-Casas et al. found previous reciprocity to be the best predictor of changes in trust for both the investor and trustee ($\rho = 0.56; \rho = 0.31$ respectively where ρ is the correlation coefficient) (King-Casas et al., 2005). Reciprocity is defined here as the fractional change of money over a round by a player in response to a fractional change of money by the player's partner. Formally, investor reciprocity on round j can be quantified as $\Delta I_j - \Delta R_{j-1}$, where ΔI_j is the fractional change in investment from the previous round, $j-1$, to the present round, j , and ΔR_{j-1} is the fractional change in repayment ($R_{j-1} - R_{j-2}$). Similarly, trustee reciprocity is quantified as $\Delta R_{j-1} - \Delta I_{j-1}$. The change in trust ΔT was thus found to be best correlated to investor reciprocity, $\Delta T_R \approx \Delta I_j - \Delta R_{j-1}$, for trustees and trustee reciprocity, $\Delta T_I \approx \Delta R_{j-1} - \Delta I_{j-1}$, for investors. Hence, by measuring these quantities of reciprocity, trust is operationalized as monetary exchange in a way that allows for online analysis of the relationship from its inception. Put another way, trust can be measured in these situations as the amount of money exchanged by each player.

The work by King-Casas et al. is important for this dissertation for several reasons (King-Casas et al., 2005). First, the research method meets the conditions for trust described in section 2.3.1. Namely, the trustor is at risk, the trustor expects that the trustee will act in a manner that mitigates his or her risk, and both parties benefit from mutual trust. Second, the research method allows for quantitative evaluation of trust in the presence of risk (loss of money) without the threat of harm to the subject. Using this method, trust is easily quantified as repayment by the trustee or investment by the

investor. King-Casas et al. intentionally minimized contextual and interactive effects by limiting interaction to remote play over a computer network (King-Casas et al., 2005). We, on the other hand, study these interactive effects by measuring responses when a human and a robot interact in situations similar to the investor-trustee game. Finally, the method originates from an interpersonal social situation and should naturally mesh with the transformation model of social interaction outlined within interdependence theory (Figure 2.5).

The King-Casas et al. research method is just one example of an entire research methodology that is largely unexplored for the purposes of human-robot interaction. Many other economic decision games exist; some requiring trust while others do not. Examples include the Ultimatum game where one player offers a division of a valuable commodity and the other player either accepts or rejects the offer for both players (e.g. Rilling, Sanfey, Aronson, Nystrom, & Cohen, 2004); the well-known prisoner's dilemma game in which both players must choose to either to cooperate for a chance at maximal reward or not to cooperate and guarantee a non-minimal reward (Axelrod, 1984). Generally, these games mesh well with interdependence theory because they share common underpinnings involving reward, cost, and outcome matrix representation. As discussed in the next chapter, this methodology will be used to explore human-robot relationship development and trust.

2.3.3 Alternative methods for evaluating trust

Most of the earliest trust research was the result of psychological and/or sociological experiments. In the past, these fields tended to rely heavily on questionnaires, observation, and interviews as a means of measuring trust (Lund, 1991; Tesch & Martin,

1983) and occasionally still do (Yamagishi, 2001). Recently, psychologists have shown an increasing willingness to use social situations, such as the prisoner's dilemma¹, as a means for controlling and measuring trust (e.g. Good, 1991; Rabbie, 1991). Sociologists have also been very active in trust research. Their work has tended to quantify trust by observing behavior within controlled laboratory experiments (e.g. Kollock, 1994; Kurban & Houser, 2005) or simulation experiments (Bainbridge et al., 1994 for a review). The work by Kollock, for example, investigated the effect of uncertainty on trust development among trading partners in a social exchange experiment (Kollock, 1994). Trust was measured in these experiments by observing the trading practices of 80 subjects and most importantly their risk-seeking and risk-averse behaviors. Specifically he explored trust with respect to an individual's commitment to another individual as measured by the equation,

$$(C_i)_t = \frac{(T_{ij} - T_{ik})^2 + (T_{ij} - T_{il})^2 + (T_{ij} - T_{im})^2 + (T_{ik} - T_{il})^2 + (T_{ik} - T_{im})^2 + (T_{il} - T_{im})^2}{3t^2}$$

where T_{ij} represents the number of trades subject i completed with subject j and t represents the number of trading periods. The variable $(C_i)_t$ signifies the commitment of individual i at time t . Overall, the equation sums the squared differences in number trades and normalizes this quantity with respect to time. Questionnaires asking each subject to rate their trust in all potential partners were also used to measure trust. He found that the degree to which the subject perceived the trading partner as trustworthy was quantifiably

¹ The Prisoner's dilemma is a well studied social situation from game theory. In the situation players decide either to cooperate or defect. The game generated a great deal of interest because rational self-interested decisions do not result in the maximal amount of reward.

related to both the risk encountered in the condition and the frequency of interaction with the partner. Kollock's work serves as an example of a quantitative mechanism developed to measure trust, but because the equations and methods appear strongly tied to Kollock's experimentation method they were not directly used in this dissertation.

Over the past decade, simulation experiments have been used as a means for trust research by both sociologists and computer scientists. Simulation offers the unique ability to control much of the external environment as well as the internal environment of the agents involved. Marsh used simulation experiments to test his early computational formulation for trust (Marsh, 1994). Marsh's work defines trust in terms of utility for a rational agent. Further, Marsh recognizes the importance of the situation and includes this factor in his formulation of trust. He estimates trust as, $T_x(y, \alpha) = U_x(\alpha) \times I_x(\alpha) \times \hat{T}_x(y)$ where $T_x(y, \alpha)$ is x 's trust in y for situation α , $U_x(\alpha)$ is the utility of α for x , $I_x(\alpha)$ is the importance of α for x , and $\hat{T}_x(y)$ is the general trust of x in y . Marsh notes many weaknesses, flaws, and inconsistencies in this formulation. For example, he states the value range he has chosen for trust, $[-1, +1)$, presents problems when trust is zero. Even so, as an early computational formulation of trust, Marsh's work is both unique and deep in its synthesis of the various psychological and sociological opinions regarding trust into a single equation. Although Marsh's research serves as inspiration, our work does not directly use Marsh's formulation.

Recently a trend in trust research has been to focus on the use of probability theory to measure and model trust. Gambetta, as mentioned in section 2.3.1, takes this approach to the extreme by equating trust to a person's probabilistic assessment of their partner's likelihood of acting in their favor (Gambetta, 1990). Josang and Lo Presti, on the other

hand, use probabilities to represent an agent's assessment of risk (Josang & Presti, 2004).

They describe an agent's decision surface with respect to risk as $F_C(p, G_S) = p^{\frac{\lambda}{G_S}}$ where C is the agent's total social capital², $F_C \in [0,1]$ is the fraction of the agent's capital it is willing to invest in a single transaction with another agent, p is the probability that the transaction will end favorably, G_S is gain resulting from the transaction and $\lambda \in [1, \infty]$ is a factor used to moderate the gain G_S . Josang and Lo Presti define reliability trust as the

value of p and decision trust as $T = \begin{cases} \frac{p - p_D}{p_D} & p < p_D \\ 0 & \text{for } p = p_D \\ \frac{p - p_D}{1 - p_D} & p > p_D \end{cases}$ where p_D is a cut-off

probability. Josang and Pope later use this model of trust to propagate trust and reputation information for the purpose of developing a secure network cluster (Josang, 2002; Josang & Pope, 2005; Josang & Presti, 2004). The work by these authors is certainly a valuable contribution to network security research. Still, it is not significantly tied in any way to interpersonal trust and assumes that the sole purpose of interaction is to propagate one's reputation. Beth et al. also use probability for the purpose of developing trust in network security claiming that the equation $v_z(p) = 1 - \alpha^p$, where p is the number of positive experiences and α is chosen to be a value high enough to produce confident estimations should be used to measure trust (Beth, Borcharding, & Klein, 1994).

² Social capital is concept from economics used to describe the value of the connections within a social network.

Castelfranchi and Falcone have been strong critics of defining trust in terms of probability because they feel this description of trust is too simplistic (Castelfranchi & Falcone, 2000). Rather, they describe a cognitive model of trust that rests on an agent's mental state. This mental state is in turn controlled by an agent's beliefs with respect to the other agent and an agent's own goals (Castelfranchi & Falcone, 2001; Falcone & Castelfranchi, 2001). Although Castelfranchi and Falcone's consideration of trust is extensive, their work has not been evaluated on any computational platform and they present no experiments. Moreover, it is not clear how their calculus would be implemented on a robot or a simulated agent.

Researchers have also explored the role of trust in machine automation. Trust in automation researchers are primarily concerned with creating automation that will allow users to develop the proper level of trust in the system. Lee and See, in an excellent review of the work in this area, note that one fundamental difference between trust in automation research and intrapersonal trust research is that automation lacks intentionality (Lee & See, 2004). Another fundamental difference is that human-automation relationships tend to be asymmetric with the human deciding how much to trust the automation but not vice versa. These fundamental differences also distinguish our work from the trust in automation research.

Many different methods for measuring and modeling trust have been explored. Trust measures have been derived from information withholding (deceit) (Prietula & Carley, 2001), agent reliability (Schillo & Funk, 1999; Schillo, Funk, & Rovatsos, 2000), agent opinion based on deceitful actions (Josang & Pope, 2005), compliance with virtual social norms (Hung, Dennis, & Robert, 2004), and compliance with an a priori set of trusted

behaviors from a case study (Luna-Reyes, Cresswell, & Richardson, 2004). Models of trust range from beta probability distributions over agent reliability (Josang & Pope, 2005), to knowledge-based formulas for trust (Luna-Reyes, Cresswell, & Richardson, 2004), to perception-specific process models for trust (Hung, Dennis, & Robert, 2004).

Table 2.2 A list of the various measures and models of trust used in previous research. The meaning of the symbols are presented within the text of this section.

Models of Trust

Author(s)	Model/Measure
(Kollock, 1994)	$(C_i)_t = \frac{(T_{ij} - T_{ik})^2 + (T_{ij} - T_{il})^2 + (T_{ij} - T_{im})^2 + (T_{ik} - T_{il})^2 + (T_{ik} - T_{im})^2 + (T_{il} - T_{im})^2}{3t^2}$
(Marsh, 1994)	$T_x(y, \alpha) = U_x(\alpha) \times I_x(\alpha) \times \hat{T}_x(y)$
(Josang & Presti, 2004)	Reliability trust $F_C(p, G_s) = p^{\frac{\lambda}{G_s}}$ decision trust $T = \begin{cases} \frac{p - p_D}{p_D} & p < p_D \\ 0 & \text{for } p = p_D \\ \frac{p - p_D}{1 - p_D} & p > p_D \end{cases}$
(Beth, Borcharding, & Klein, 1994)	$v_z(p) = 1 - \alpha^p$
(King-Casas et al., 2005)	$\Delta T_R \approx \Delta I_j - \Delta R_{j-1} \text{ and } \Delta T_I \approx \Delta R_{j-1} - \Delta I_{j-1}$

Often these measures and models of trust are tailored to the researcher's particular domain of investigation. Luna-Reyes et al., for example, derive their model from a longitudinal case study of an interorganizational information technology project in New York State (Luna-Reyes, Cresswell, & Richardson, 2004). This model is then tested to ensure that it behaves in a manner that intuitively reflects the phenomena of trust. A review of computational trust and reputation models by Sabater and Sierra state, "... current (trust and reputation) models are focused on specific scenarios with very delimited tasks to be performed by the agents" and "A plethora of computational trust and reputation models have appeared in the last years, each one with its own characteristics and using different technical solutions (Sabater & Sierra, 2005)."

The alternative methods for evaluating trust discussed in this section highlight a diversity of approaches and domains the topic of trust touches on. Table 2.2 lists several methods of evaluating trust proposed by different authors. It is our belief that by relating these many models to a unifying framework for social situations we will lend insight and progress to this nascent field. Chapter 8 delineates our method for recognizing and acting in situations requiring trust.

2.4 Summary

To summarize, interdependence theory provides a theoretical basis for representing relationships and social situations in the proposed dissertation (Kelley, 1979; Kelley et al., 2003; Kelley & Thibaut, 1978; Rusbult & Van Lange, 2003). It offers a basis for understanding the social actions of a human; it also provides a framework for determining the proper robotic responses to a given social situation; finally, interdependence theory forms a “conceptual skeleton” used to describe social situations some of which involve trust (Kelley et al., 2003).

Because interdependence theory provides a general means for representing social situations which is not tied to a particular environment or paradigm, it is possible to segregate those situations that demand trust from those that do not without altering the interdependence framework. We defined trust as *a belief, held by the trustor, that the trustee will act in a manner that mitigates the trustor’s risk in a situation in which the trustor has put its outcomes at risk*. This definition is derived from a definition offered by Lee and See.

An experimental methodology involving economic decision games, similar to those used by King-Casas et al (King-Casas et al., 2005), has been detailed as a means for

investigating human-robot interaction. These games generally allow for iterative interactions, use the potential loss of a valued commodity as risk, and are capable of quantifying an individual's actions in terms of trust. Moreover, these games have a long and established history as a means for exploring interaction.

Several methods of social learning have also been discussed. These include learning of associations, the use of diagnostic situations for credit-assignment and stereotyping. Stereotyping is a process by which interpersonal schema relating perceptual features to distinctive clusters of traits are used to bootstrap understanding of a novel partner.

This chapter has reviewed literature covering several general areas of research relevant to this dissertation. The goal has been to highlight the connection of these areas to the principal and subsidiary research questions in a clear and coherent manner. The next chapter introduces our methodology for investigating these topics.

CHAPTER 3

A METHODOLOGY FOR INVESTIGATING THE THEORY

UNDERLYING HUMAN-ROBOT INTERACTION

This chapter presents a methodology for investigating the theory that underlies human-robot interaction. The aim is to introduce the reader to several new methods of human-robot interaction research developed for this dissertation. As will be shown, these methods are particularly applicable to the creation of a general, principled framework for human-robot interaction. As detailed in section 2.1, current human-robot interaction experimental methods often focus on feasibility studies, usability studies, and in-field experiments (Fong, Nourbakhsh, & Dautenhahn, 2003). Several researchers note that these methods are generally inadequate for theory research because the terms used are not defined, reproducibility is rarely possible, and theories are either absent or not falsifiable (Bethel & Murphy, 2008; Feil-Seifer, 2008; Heckel & Smart, 2008).

3.1 A Method for HRI Theory Research

Theories are developed in order to explain or better understand a natural phenomenon. According to Philip Kitcher, scientific theory should 1) “open(s) up new areas of research”, 2) “consist of just one problem-solving strategy, or a small family of problem-solving strategies, that can be applied to a wide range of problems”, 3) be “testable independently of the particular problem” and 4) must be falsifiable (Kitcher, 1982).

This dissertation studies uses a robot to study the natural phenomena of social interaction. It is common to think of social interaction as existing outside the realm of

robotics. We do not believe that this is the case. Rather, we prescribe to the belief that social interaction is a critical component of intelligence in general (Humphrey, 1976). Roboticists have long sought to make robots more intelligent (e.g. Brooks, 1991; MacFarland & Bosser, 1993). Hence, at the highest level, we explore social interaction as a means to make robots more intelligent. In a practical sense, the use of a robot allows us to explore social interaction in a manner never before attempted. Rather than using indirect mechanisms to infer a human's psychological representation of its interactive partner and social situation, the use of a robot allows us to directly examine this representation. In the end, the results we obtain should produce both better social control algorithms for the robot and, possibly, a stronger connection of psychological theory to its perceptual and behavioral underpinnings. Still, this dissertation does not directly investigate human psychology. Rather, we focus on how changes in the social environment produce representational and behavioral changes in a robot. Our metrics therefore measure changes with respect to the robot, not the human. As detailed in the next section, this means that human behavior is a controlled variable.

Our overarching theory is that

social interaction results in outcomes for each individual, that these outcomes must be represented in order to reason about future interactions including the development of a relationship, and that the representation of these outcomes affords a robot the ability to reason about other social phenomena, such as trust.

This statement paraphrases the research questions posed in chapter one. Referring back to Kitchner's criteria for a scientific theory, our theory does open new areas of investigation specifically by developing novel algorithms that should allow a robot to interact in a wider variety of environments; our theory presents a small family of problem solving techniques, centering on the use of the outcome matrix, that are applicable to a large body of problems; our theory is general, not tied to a specific problem or environment; and finally, by showing that outcome matrices cannot be produced from perceptual information or that these matrices do not result in improved social behavior by the robot, our theory is falsifiable.

Our method for studying this theory is to follow a precise series of steps. First, we define all terms for the particular phenomena being studied (social interaction, trust, or relationships for example). These definitions may or may not result in particular assumptions on which our results will rest. If so, then these assumptions are explicitly stated. Next, given the definitions, we systematically develop representations, algorithms, and/or corollaries to our original theory. Finally, these representations, algorithms and corollaries are tested using a particular experimental paradigm. It is important to note that the experimental results will not "prove" our original theory. Rather they simply lend support to our original argument. Proof of a theory can only be gained by independent confirmation from other researchers (Popper, 1963).

Social interaction is governed by three variables: 1) the first interacting individual; 2) the second interacting individual; and 3) the environment (Rusbult & Van Lange, 2003). In human-robot interaction either the first or the second individual is a robot and the remaining individual is a human. As mentioned, the purpose of this dissertation is to study

and develop techniques that will allow a robot to interact. If we are to be successful, then it is useful to control the remaining two variables related to interaction. Namely, we should attempt to control for the behavior of the human and for the environment. In the sections that follow, we present several methods that allow us to this.

3.2 Controlling Human Behavior—Actor scripts

It is helpful, when evaluating a robot's ability interact, if the behavior of the robot's human partner is controlled for. As experimenters, control of the human's behavior allows us to focus our investigation on a single dependent variable—the resulting actions of the robot—rather than having to infer the reasons for the robot's behavior. This strict control is helpful during the early stages of experimentation and theory development because it allows us to quickly rule incorrect theories or faulty algorithms. Later in the experimental cycle, we can loosen our control to over the human's behavior to allow for more realistic pseudo-random behavior on the part of the human. In a sense, stress testing our algorithms and theories before introduction in to real world environments.

Laboratory experiments involving controlled human behavior are standard in many psychology experiments (e.g. Milgram, 1974). These experiments typically require that the experimenter's confederate follow a predefined script often acting the part of a fellow subject. This script explains how the actor should behave in all of the situations he or she will face. In much the same way, our evaluation of the robot's interactive behavior requires that the human partner act in a scripted manner. We use the term *actor script* to describe a predefined set of interactive instructions that the human will follow when interacting with the robot. Actor scripts are used in several of the experiments conducted as part of this dissertation.

Actor scripts are methodological in nature and are a contribution of this dissertation. An actor script is created by first delineating the situations that the human-robot dyad will encounter. Once the situations have been determined, the human's actions can be dictated in several different ways. For example, one could develop scripts related to the actions and preferences of a firefighter in a search and rescue environment. These scripts could be generated by observing real firefighters in search and rescue environments or, possibly, by constructing the script from data and information related to search and rescue. Another possibility is to assign the human a broad social character (see Appendix C for examples) and to then select actions in accordance with the assigned character. For example, if the human is assigned the social character of egoist then the human will select the outcome matrix action that most favors his or her own outcomes. To complete the actor script, actions are determined for each interaction, possibly being contingent on the robot's prior behavior, and a list or flowchart is created that the human follows when interacting with the robot.

3.3 Controlling the Environment—Social Situations

The use of games and predefined social situations as a method for exploring human interactive behavior was discussed in section 2.2.4. To briefly review, social psychologists and neuroscientists have begun using games such as the Investor-trustee game and Ultimatum game to investigate how and why humans interact (Sanfey, 2007). These simplistic games place interacting individuals in controlled social situations which allow researchers to tease apart the impact of different factors on interactive behavior. The Ultimatum game, for example, forces one individual to offer a division of a valuable commodity and the other individual either accepts or rejects the offer for both players

(e.g. Rilling, Sanfey, Aronson, Nystrom, & Cohen, 2004). Ethnographic results show that humans will routinely reject proposals in which they receive less than 20% (Henrich et al., 2003). It is speculated that humans consider and value the fairness of a proposal and that this preference for fairness often supersedes the money that would have been received by accepting an unfair proposal.

The Ultimatum game is not merely an academic exercise. Rather, the characteristics of this social situation occur daily in many routine social interactions. For example, the Ultimatum game is often manifest in simple negotiations, such as determining how much work an individual will do for a predetermined pay. Because the characteristics of this game, and many others, are normal components of everyday human social interaction, it is important that robots recognize and master situations such as these.

Hence, one potential method for studying the theory underlying human-robot interaction is to place robots in controlled social situations such as the Ultimatum game. These situations afford control over the external environmental factors which could influence the robot's decision making and model construction processes.

Randomly generated social situations can also be used to test the generality of a theory. Social situations are randomly generated if the outcome values that comprise the situation are randomly created and nominal actions are assigned. Randomly generated situations may originate from any location in the interdependence space (Figure 2.3). Thus, by testing the robot's response to many randomly generated social situations, one can garner evidence supporting a theory with respect to all possible social situations, rather than one social situation in particular. We use the term numerical simulation to describe experiments which employ a large number of randomly generated social

situations. For example, given a theory of trust, we could use the investor-trustee game to test whether or not the robot has recognized that trust will effect its partner's decision in the laboratory. Once our theory of trust has been validated in a social situation commonly agreed to involve trust we then use randomly generated social situations to test our model over the entire interdependence space.

This dissertation employs both of these research methods, occasionally validating theories with particular social situations and later using randomly generated social situations to expand these original results. The sections that follow describe the methods of evaluation particular to this dissertation.

3.4 Evaluative Methods

As mentioned in section 1.1, all experiments involve interaction between a single robot and a single human. Some experiments require that each individual select an interactive action simultaneously while others demand a series of actions being selected. The actions available depend on the experimental condition, environment, and so on. After selecting an action, both the robot and the human perform the action. Once the action had been performed, the human tells the robot the value of the outcome that he or she had received as a result of both actions being selected. The robot and the human continue taking turns until the experiment is complete.

3.4.1 Numerical simulation experiments

We conduct two broad types of simulation experiments as part of this dissertation: numerical simulations and simulations within a simulation environment. Numerical simulations of interaction focus on the quantitative results of the algorithms and

processes under examination and attempt to simulate aspects of the robot, the human, or the environment. As such, this technique offers advantages and disadvantages as a means for discovery. One advantage of a numerical simulation experiment is that a proposed algorithm can be tested on thousands of outcome matrices representing thousands of social situations. One disadvantage of a numerical simulation is that, because it is not tied to a particular robot, robot's actions, human, human's actions, or environment, the results, while extremely general, have not been shown to be true for any existing social situation, robot, or human.

Our numerical simulations of interaction typically simulated both the decisions and action selection of the human and the robot. Actions in these types of simulations are nominal and do not represent actual actions performed in the environment. These nominal actions are grounded by the rewards and costs the robot receives when selecting them, regardless of the mechanics of the actions performance. Moreover, numerical simulations do not utilize an interactive environment outside of the outcome matrix itself. Hence, the domain, task, and physics of the world are abstracted away in these types of simulations.

To understand the role and value that such simulations can play in the science and exploration of human-robot interaction, consider the following question, "what percentage of situations warrant deception?" This question is important, because if the percentage of situations warranting deception is very small, then, perhaps, the study of deception itself is unjustified. Still, a systematic examination of all possible grounded social situations within a given scenario and environment seems infeasible. We can, however, create outcome matrices representing random situations that the robot could

encounter, then allow the simulated robot and a simulated human partner to select actions, and record the outcomes that result. This type of experiment allows one to rapidly explore the space of social situations to better understand aspects of interaction such as deception or trust.

3.4.2 Simulation experiments within a simulation environment

Many of the simulation experiments conducted for this dissertation utilize USARSim, a collection of robot models, tools, and environments for developing and testing search and rescue algorithms in high-fidelity simulations (Carpin, Wang, Lewis, Birk, & Jacoff, 2005). USARSim's robot models have been shown to realistically simulate actual robots in the same environment (Wang, Lewis, Hughes, Koes, & Carpin, 2005). Moreover, USARSim provides support for sensor and camera models that allow a user to simulate perceptual information in a realistic manner. USARSim is freely available online.

USARSim is built on Epic's Unreal Tournament (UT) game engine. A license for the game engine costs approximately five dollars. Unreal Tournament is a popular 3D first person shooter game. Unreal Tournament's game engine produces a high-quality graphical simulation environment that includes the kinematics and dynamics of the environment. Numerous tools for the creation of new environments, objects, and characters are included with the game. These tools can be used to rapidly prototype novel environments at minimal cost. Moreover, several complete environments and decorative objects are freely available online.



Figure 3.1 Screenshots from the simulation environment are depicted above. The top left shows a household environment. The top right depicts a museum environment. The bottom left illustrates the assistive environment. The bottom right illustrates the prison environment.

Table 3.1 List of colored objects in each environment.

Colored Objects		
Object	Color	Environment
Biohazard	green	Search and rescue
Fire	red	Search and rescue/Museum
Victim	yellow	Search and rescue
Patient	light blue	Assistive
Medicine	light blue	Household
Homeowner	green	Household
Intruder	dark blue	Household/Museum
Prisoner	purple	Prison
Visitor	light blue	Prison

Figure 3.1 depicts examples of environments we created for this dissertation using Unreal Tournament tools. The household environment modeled a small studio apartment and contains couches, a bed, a television, etc. (Figure 3.1 top left). The museum environment models a small art and sculpture gallery and contains paintings, statues, and exhibits (Figure 3.1 top right). The assistive environment models a small hospital or physical therapy area and contains equipment for physical, art, music and occupational therapy (Figure 3.1 bottom left). The prison environment models a small prison and contains weapons, visiting areas, and a guard station (Figure 3.1 bottom right). Finally, the search and rescue environment models a disaster area and contains debris fields, small

fires, victims, and a triage area (Figure 3.2 and Figure 3.4). Some objects in these environments were colored to aid the robot's recognition of these items. Table 3.1 lists objects that were artificially colored.

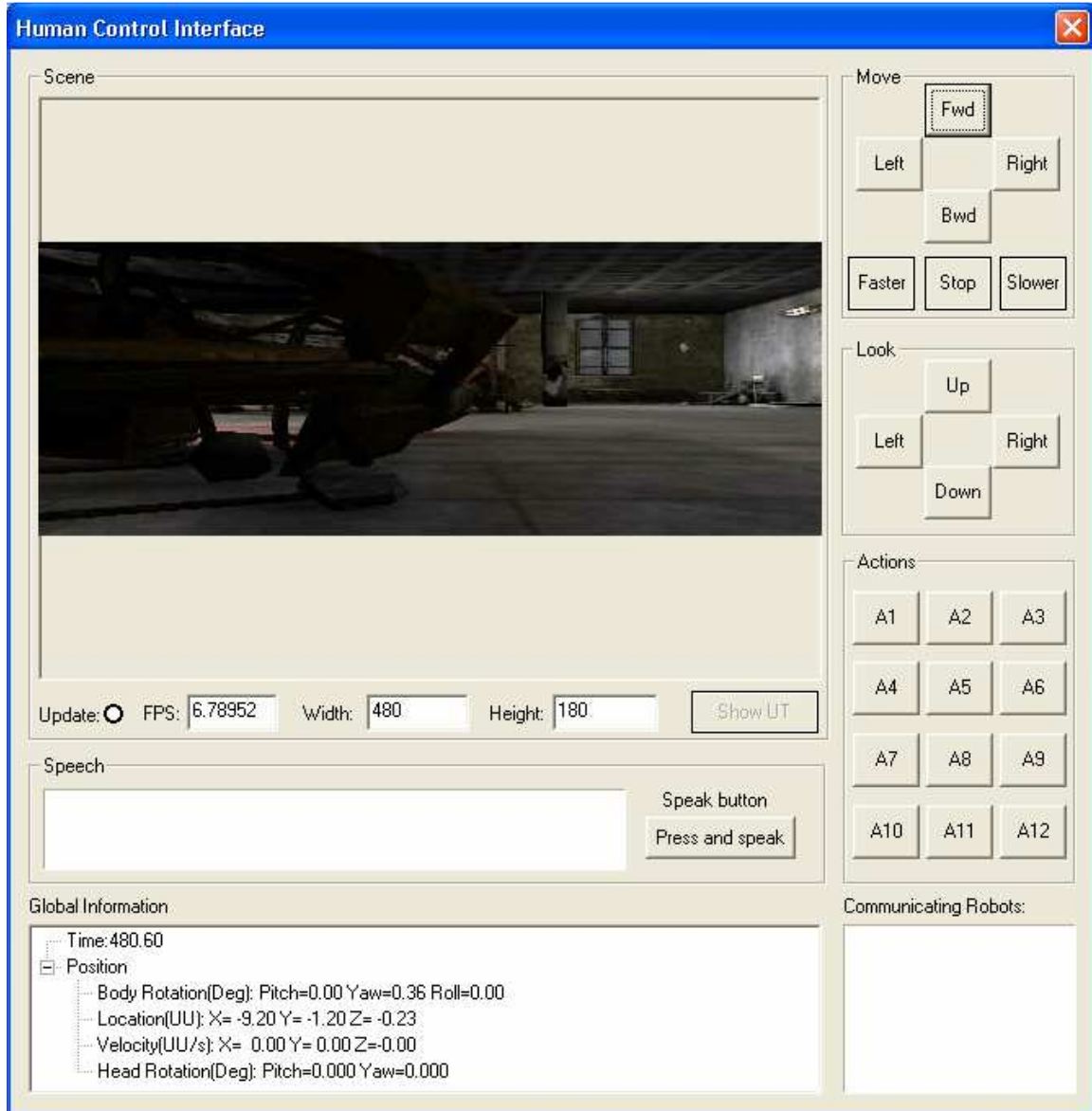


Figure 3.2 The interface used by the human to move and interact in the simulated environment. The environment shown is the search and rescue environment.

We developed a software interface that allows a human to interact with the robot in the simulation environment (Figure 3.2). This interface was developed from an existing USARSim tool (Zaratti, Fratarcangeli, & Iocchi, 2006). Using the interface the human

can move and look around the environment. The human can also speak to the robot using a predefined grammar of commands and can hear the robot's responses.

The simulation environment employs three computers running in concert (Figure 3.3). The simulation server runs the USARSim simulation engine and serves video and position data to clients. The robot and the human connect to the simulation server as clients. The robot client controls the robot's behavior within the simulation. The human control interface allows the human to move, look, and perform actions within the simulation environment. The human control interface also acts as a speech server translating speech into strings for the robot client and strings into synthesized speech for the human.

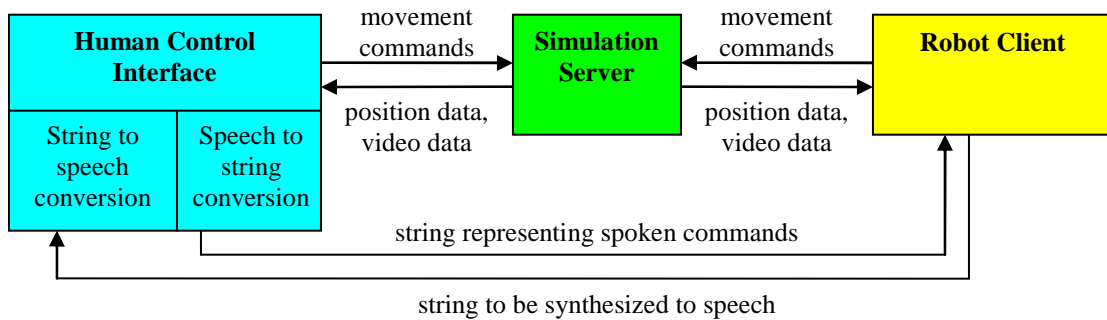


Figure 3.3 Depiction of the network and control setup used to perform simulation experiments. The human interacts through the Human Control Interface. The simulation server runs the simulation environment and feeds information to both the Human Control Interface and the Robot Client.



Figure 3.4 The figure depicts a split screen view of the search and rescue environment. The human sees only the top half of the figure. The bottom half shows the robot situated in the environment.

3.4.3 Laboratory experiments

In contrast to the experiments conducted in simulation, several experiments were conducted in the mobile robot lab and used a real robot. The experimental area in the lab was modeled after a search and rescue or maze style environment (Figure 3.5). The environment included mock victims and hazard signs.



Figure 3.5 An overview of the maze environment used as a mockup of a simple search and rescue environment. One corner of the maze had two dolls representing children (center photo) and the other corner had a simulated fire and biohazard (right photo).

3.4.4 Evaluation platforms

Two types of robots were used in the experiments. We used a Pioneer DX with a front mounted camera for the experiments involving partner modeling and stereotypes (Chapter 5). Figure 3.4 depicts the robot operating in a simulated search and rescue environment and Figure 3.6 shows the real robot platform in the laboratory. This robot is a wheeled robot with mobility over smooth flat surfaces. The robot's environment supports independent locomotion over short distances, approximately 15 meters, which was far enough for all experiments.

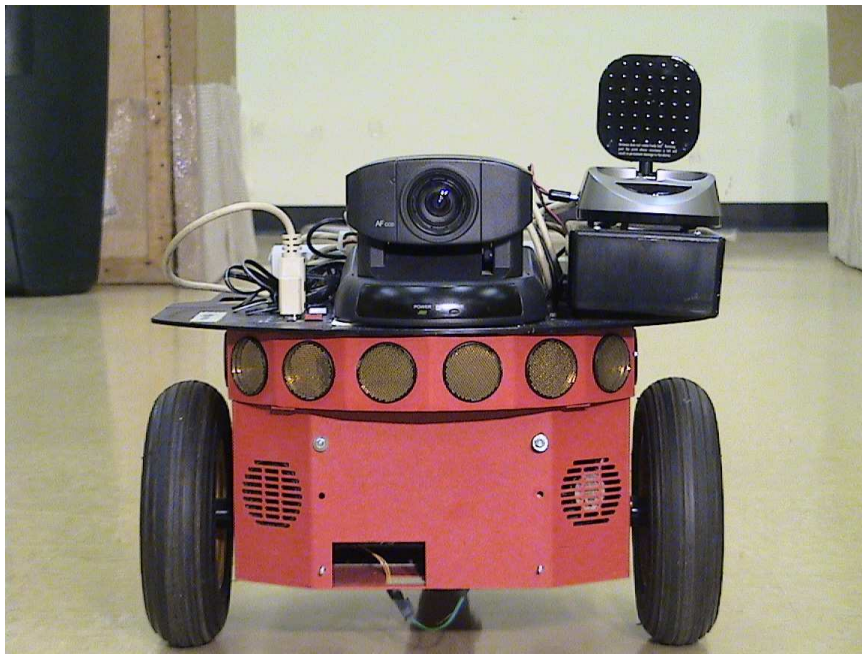


Figure 3.6 A photo of the Pioneer DX used to perform the laboratory experiments.

The robot's camera is a 320 by 240 pixel video camera mounted on the front of its flat deck. The effective frame rate of this video camera is approximately 10 frames per second. The computational platform used for control, perceptual processing, and the test algorithms was a standard Toshiba Satellite laptop. An internal wireless LAN card was used to transmit and receive information from the robot.

WowWee's Rovio robot was used in our experiments exploring the robot's ability to select the most trusted partner (Chapter 8). This robot is also a wheeled robot with mobility over smooth flat surfaces. The robot's camera is a 640 by 480 pixel webcam mounted on the top of the robot's extendable neck. Communication with the robot is accomplished via the robot's wireless network card. The Rovio comes with a docking station and infrared beacon easing the robot's navigation task back to the docking station.



Figure 3.7 A photo of the Rovio mobile robot. The robot's neck is point towards the bottom of the image and is in the unextended position. The webcam is at the end of the neck.

3.4.5 Interactive communication

The human and the robot used speech to communicate. The robot used speech synthesis to communicate questions and information to the human partner. Speech recognition translated the spoken information provided by the human. Microsoft's Speech SDK provided the speech synthesis and recognition capabilities (Microsoft Speech SDK 5.1 information page, 2006; Open Mind Speech, 2006).

3.4.6 Human partners

Interaction involved a single robot and a single human. The robot's human partner was assigned a predetermined list of perceptual features that were used by the robot for identification or as evidence of the partner's type. The human's perceptual features were spoken to the robot. Table 3.2 lists all partner features.

Table 3.2 A list of partner features is presented above. Several of the features were devised because of their notional significance.

Partner Features	
Feature Name	Values
Gender	<man, woman>
Height	<tall, medium, short>
Age	<young, middling, old>
Weight	<heavy, average, thin>
Hair color	<blonde, black, brown, red>
Eye color	<blue, green, brown>
Tool 1	<axe, gun, stethoscope, baseball-cap>
Tool 2	<oxygen-mask, badge, medical-kit, backpack>

The human's actions were scripted. In other words, the human selected from a predefined series of actions that were contingent on the robot's prior actions and the experimental condition. Section 3.2 has already discussed the use of controlled human behavior in the experiments. Because the experiments controlled for the human's features and actions, all experiments could be conducted by a single human partner. Still, three different operators were used (a 20 year old American woman, a 20 year old Indian-American woman, and a 33 year old American male) to rule out any possibility of experimenter bias. When interacting in the same environment with each different operator the outcome matrices created by the robot were identical.

3.4.7 Perceptual underpinnings

We used a combination of computer vision and speech recognition for much of this work, including the recognition of interactive partners, navigation, and object detection and recognition.

Partner Recognition

The recognition of interactive partners is critical for this work. The robot queries the human for the human's perceptual features and then matches these features to a preexisting model of the partner. Features were spoken. These features were used both to retrieve models of known partners and to construct identities for unknown partners. Table 3.3 summarizes the perceptual infrastructure used.

Table 3.3 A summary of the perceptual requirements, the software package used, and their usage.

Perceptual Infrastructure Summary		
Necessity	Software Package	Usage
Partner Recognition	Microsoft Speech SDK	Used to communicate partner features and outcome values received by human partner.
Object Recognition	OpenCV	Used to recognize specific objects in the environment, determine environment type.
Navigation/Localization	OpenCV	Used to aid the robot in navigating over short distances in an indoor environment.

Object Recognition and Navigation

Some experiments required that the robot search for and locate objects. The objects were not occluded and adequately lit. Searching for objects requires rudimentary navigation. All of the experimental environments were passable by the robot. To aid navigation, in simulation the robot received accurate feedback of its location. Color blobs were used to denote objects. Objects were color coded for recognition purposes (Table 3.1). Laboratory experiments used visual landmarks to provide location feedback (Figure 3.8).



Figure 3.8 In laboratory experiments the robot uses landmark detection to aid navigation.

3.4.8 Robot behaviors

Many of the robot's actions were related either to the performance of relatively simple search and navigation tasks or to interactive communication. Table 3.4 summarizes the requisite behaviors and their purpose.

Capture Behaviors

Capture behaviors were used to determine the type of environment, partner, or the result of an interaction. The `CaptureEnvironmentFeatures` behavior uses the robot's camera to gather information about the robot's operational environment. The `CapturePartnerFeatures` behavior asks the human to state their features which are then saved by the robot. The `CaptureInteraction` behavior asks the human to state the outcome they received after performing an action in the environment.

Table 3.4 The table provides a summary of the behaviors used.

Behavior	Purpose
CaptureEnvironmentFeatures	The robot uses its camera to determine the type of environment.
CapturePartnerFeatures	The robot asks the human to state their features
CaptureInteraction	The robot asks the human to state the value of outcome they received.
Speak-X	The robot states X
SearchFor-X	The robot navigates from waypoint to waypoint scanning its camera in search of object X.
GuideTo-X	The robot requests that the human follows and then navigates to position X.
Observe-X	The robot moves to position X, positions its camera and remains.
Light-X	The robot moves to position X and turns on its light.

Functional Behaviors

Functional behaviors allowed the robot to do or say things within the simulated or real environment. Functional behaviors require additional knowledge in the form of a phrase, environment location, or object type. In all experiments the robot was given the knowledge of stock phrases, the location of objects, or available objects in an environment. For example, the `SearchFor` behavior takes as a parameter an object such as *fire*. The robot has been programmed with information that fire is red. The robot will then search for a red object in the environment. When it finds the object it relays the position of the located object to the human.

3.5 Example Interaction

For clarity, we will briefly overview the process that occurs during a typical interaction with a human partner. When the robot is powered up, it first uses the `CaptureEnvironmentFeatures` to determine the type of environment. In simulation, it uses its camera to take an image of the environment and compares the image to images of different environment types. It selects the environment type which most closely

matches the camera image. Once it knows the environment type, it determines which of its actions are appropriate for the environment (this will be discussed in greater detail in chapter 5). Next it uses the `CapturePartnerFeatures` behavior to gather information about its interactive partner. Once it has the partner's features, it constructs an outcome matrix (detailed in chapter 5), uses the matrix to select an action, and performs the action. After the action has been performed the robot returns to a predetermined location to interact again.

This chapter has presented a methodology for investigating the theory that underlies human-robot interaction. We contribute several methods novel to human-robot interaction research. These methods allow one to control for the behavior of the interacting human and to control the social situation. We have also described a simulation environment capable of high-fidelity simulations in naturalistic environments using a variety of robot models. Coming back to our principal research question, we believe these methods will allow for a systematic investigation of the our overarching theory—that interaction results in outcome, that these outcomes must be represented in order to develop a relationship, and that the representation of these outcomes affords a robot the ability to reason about trust. The chapter that follows presents our framework for representing and reasoning about human-robot interactions.

CHAPTER 4

A FRAMEWORK FOR REPRESENTING AND REASONING ABOUT HUMAN ROBOT SOCIAL INTERACTION

This chapter presents a framework by which a robot can represent and reason about its interactions. Our framework draws heavily from interdependence theory, a social psychological theory of human relationship development (Kelley & Thibaut, 1978). The previous chapter presented a methodology for human-robot interaction research. In this chapter we begin to put this methodology to use by defining the terms and concepts that will form the core of our framework. The chapter begins with social interaction, arguably the most fundamental concept in human-robot interaction.

4.1 Defining the term Social Interaction for Robots

The term social interaction is often used by human-robot interaction researchers (Rogers & Murphy, 2001). But what do we mean by social interaction? The first subsidiary question posed in chapter one proposes an exploration of this question. We begin with an established definition from psychology.

Social interaction has been defined as influence—verbal, physical, or emotional—by one individual on another (Sears, Peplau, & Taylor, 1991). This definition is a broad one, potentially encompassing phenomena such as gossip, online interaction, and reputation. A more narrow definition of social interaction is proffered by Goffman, who, on the other hand, defines it as face-to-face behavior occurring within a defined social situation (Goffman, 1959). Because the definition by Sears, Peplau, and Taylor covers a broader spectrum of phenomena, we have chosen it as our running definition for the term social interaction. Namely,

social interaction is *influence—verbal, physical or emotional—by one individual on another. (Sears, Peplau, & Taylor, 1991).*

For the remainder of this dissertation a specific social interaction is termed an interaction.

We use the definition to reason about the type of information that should be presented in a representation of interaction. The definition centers on the influence individuals have on one another. This influence can be represented as a real number. Thus real numbers representing each individual's influence on the other individual should be present in our representation. The definition also implies that during social interaction individuals actively deliberate over and select actions, which in turn influence their interactive partner. Hence, our representation must also include information about the actions each individual is considering. Moreover, for each pair of actions we must represent the influence that a selection of the action would have on each individual. Finally, our representation must include information about who is interacting.

Outcome matrices contain all of this information. An outcome matrix not only identifies the individuals interacting but also contains information about the actions available to both individuals and the influence that results from the selection of each pair of actions. The layout of an outcome matrix is depicted in Figure 2.4. As has been mentioned, the outcome matrix has a long history as a representation for interaction in a variety of different fields (Chadwick-Jones, 1976).

Finally, we note that in this context the term action is used to describe any mechanism by which one individual influences their environment, including other individuals within that environment. Much in the same way that action is defined in reinforcement learning, an action here can be a low-level control or a high-level behavior (Sutton & Barto, 1998). Hence an action could be a spoken phrase, a performed behavior, or even a collection of behaviors.

4.1.1 Social situations and interaction

Outcome matrices are also used to represent social situations. The term situation has several definitions. The most apropos for this work is “a particular set of circumstances existing in a particular place or at a particular time (Situation, 2007).” A working definition for social situation, then, is a situation involving more than one individual where an individual is defined as either a human or a social robot. Put another way, a social situation characterizes the environmental factors, outside of the individuals themselves and their actions, which influence interactive behavior. In other words, a social situation describes the social context surrounding an interaction between individuals (Rusbult & Van Lange, 2003).

A social situation is abstract, detailing the general environment of the interaction. An interaction, on the other hand, is concrete with respect to the two or more individuals and the social actions available to each individual. For example, the prisoner’s dilemma describes a particular type of social situation. As such, it can, and has been, instantiated in numerous different particular social environments ranging from bank robberies to the front lines of World War I (Axelrod, 1984). Hence, the term interaction describes a discrete event in which two or more individuals select particular interactive behaviors as part of a social situation or social environment.

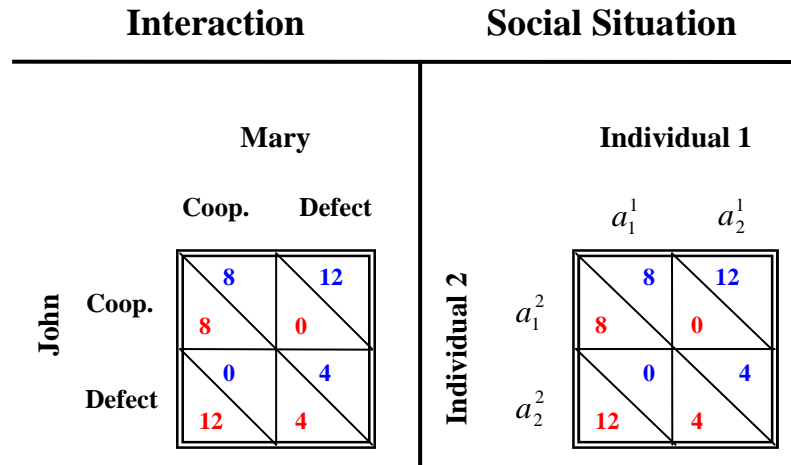


Figure 4.1 This figure depicts the difference between an interaction and a social situation. Both outcome matrices above depict the Prisoner’s dilemma. The left most matrix depicts the prisoner’s dilemma as an interaction between two specific people selecting between specific actions. The right most matrix depicts the prisoner’s dilemma as an abstract social situation, without specific actions or individuals.

A social situation may or may not proffer interaction. Interdependence theorists state that interaction is a function of the individuals interacting and of the social situation (Rusbult & Van Lange, 2003). Technically, social situations marked by no interdependence on the part of either individual afford no interaction because the individuals do not influence one another. Although a social situation may not afford interaction, all interactions occur within some social situation. Figure 4.1 graphically depicts the difference between a social situation and an interaction. Interdependence theory represents social situations involving interpersonal interaction as outcome matrices.

4.1.2 A formal notation for describing human-robot interaction

In this section, we use the definitions from the previous section to create formal notation for describing human-robot interaction. This notation builds from our use of the outcome matrix as a means for representing interaction and social situations.

As a representation, interdependence theory's outcome matrices are equivalent to game theory's normal form game (Figure 2.4 and Figure 4.1). The normal form representation of a game consists of 1) a finite set N of players; 2) for each player $i \in N$ a nonempty set A^i of actions; 3) the payoff obtained by each player for each combination of actions that could have been selected (Gibbons, 1992). Let $a_j^i \in A^i$ be an arbitrary action j from player i 's set of actions. Let (a_1^1, \dots, a_k^N) denote a combination of actions, one for each player, and let u^i denote player i 's payoff function: $u^i(a_1^1, \dots, a_k^N) \rightarrow \mathfrak{R}$ is the payoff received by player i if the player's choose the actions (a_1^1, \dots, a_k^N) .

The terminology employed when discussing the outcome matrices that describe a social situation occasionally differs from some of the terms of game theory. Game theory considers the actions of players whereas interdependence theory considers the actions of actors or individuals. As stated in chapter one, we use the term individual to denote either a human or a social robot. The reward obtained when players select actions is a payoff in game theory and an outcome in interdependence theory. Payoff functions determine the value of these payoffs in game theory whereas utility functions determine the value of outcomes. For the most part, the differences in terms are simply different names for the same thing. One difference in representation, however, is game theory's use of strategies. A strategy in game theory describes a complete plan of action that a player will take. Because game theory assumes that all players are rational, each player is bound to its strategy and the normal form game can be defined in terms of strategies rather than actions. Interdependence theory does not assume rationality and hence does not describe the outcome matrix with respect to strategies. Throughout this dissertation the interdependence theory terminology will be used.

Returning to our notation, subscripts denote order and superscripts denote ownership. Individual 1 will always be used to describe the individual listed above the outcome matrix, when the outcome matrix is depicted graphically (Figure 4.1). Without loss of generality, the robot will always be depicted as individual 1. Individual 2 is always the robot's human partner and the individual listed to the left of a graphically depicted matrix (Figure 4.1). Thus, action a_2^1 denotes individual 1's second action, and action a_1^2 denotes individual 2's first action. The term o denotes an outcome value within the matrix. The superscripts and subscripts for all outcome values are depicted in Figure 2.4. The left hand subscripts can be used to reflect the actions selected by each individual, with individual 1's action first. Right hand subscripts denote order, for example $o_1^1 > o_2^1$ indicates that individual 1's outcome has increased. Again, right hand superscripts denote the individual. Game theory also uses the superscript i and $-i$ to abstractly represent an individual and their interactive partner. Hence, individual 1's first action can also be represented as a_1^i . The first action of individual 1's partner is also expressed as a_1^{-i} . The term O is used to denote an outcome matrix. A particular outcome within a matrix can also be expressed as a function of an outcome matrix and an action pair, thus $O^1(a_2^1, a_1^2) =_{12} o^1$ and $O^2(a_2^1, a_1^2) =_{12} o^2$. Here the variable o denotes an outcome value. The term $_{12}o^2$ denotes that it is the second individual's outcome from the first row and second column of the matrix. The temporal order of action selection is expressed as $i \Rightarrow -i$ if individual i acts before individual $-i$ and $i \Leftrightarrow -i$ if both individuals act at the same time. For example, individual 1 acting before individual 2 is expressed as $1 \Rightarrow 2$.

Figure 4.2 demonstrates the use of this notation in an actual HRI experiment from German Research Center for Artificial Intelligence (Zender, Mozos, & Jensfelt, 2007). In

this experiment, an exploring robot asks a human assistant for information about the environment. It asks the human whether or not a door is present. The human states “no”, but the robot fails to recognize the response. The human repeats “no” ten times finally stating “there is no f####ing door here.” The robot recognizes this final response and proceeds.

4.2 Partner Modeling

Several researchers have explored how humans develop mental models of robots (e.g. Powers & Kiesler, 2006). A mental model is a term used to describe a person’s concept of how something in the world works (Norman, 1983). We use the term partner model (denoted m^{-i}) to describe a robot’s mental model of its interactive human partner. We use the term self model (denoted m^i) to describe the robot’s mental model of itself. Again, the superscript $-i$ is used to express individual i ’s partner (Osborne & Rubinstein, 1994).

Demonstration of outcome matrix notation used to describe an HRI Experiment

a_1^1 = Ask if there is a door. a_1^2 = State no.
 a_2^1 = Thank and move forward. a_2^2 = State yes.
 a_3^1 = No response.



We assume $o_{10}^2 > o_1^2$ based on the laughing in the video.

#	Dialog	Notation for Robot	Notation for Human
1	R: Is there a door there?	$O^1(a_1^1)$	$O^2(\phi)$
2	H: No.	$O^1(\phi)$	$O^2(a_3^1, a_1^2) = o_1^2$
3	H: No.	$O^1(\phi)$	$O^2(a_3^1, a_1^2) = o_2^2 < o_1^2$
4	H: No.	$O^1(\phi)$	$O^2(a_3^1, a_1^2) = o_3^2 < o_2^2$
5	H: No.	$O^1(\phi)$	$O^2(a_3^1, a_1^2) = o_4^2 < o_3^2$
6	H: Robot no.	$O^1(\phi)$	$O^2(a_3^1, a_1^2) = o_5^2 < o_4^2$
7	H: Robot no.	$O^1(\phi)$	$O^2(a_3^1, a_1^2) = o_6^2 < o_5^2$
8	H: No.	$O^1(\phi)$	$O^2(a_3^1, a_1^2) = o_7^2 < o_6^2$
9	H: No.	$O^1(\phi)$	$O^2(a_3^1, a_1^2) = o_8^2 < o_7^2$
10	H: Robot No. There is no f####ing door here.	$O^1(\phi)$	$O^2(a_3^1, a_1^2) = o_9^2 < o_8^2$
11	R: Ok, thank you for helping me.	$O^1(a_2^1)$	$O^2(a_2^1, a_1^2) = o_{10}^2 > o_1^2$

Figure 4.2 The figure above demonstrates the use of our outcome matrix notation in a human-robot interaction experiment conducted by German Research Center for Artificial Intelligence (Zender, Mozos, & Jensfelt, 2007). The robot (pictured in the top right photo) asks the human whether or not a door is present. The human says no 10 times before the robot responds. Notation is provided for the robot and the human. The human's outcomes decrease every time he must repeat the command. Finally, when the robot responds, the human's outcomes are increase dramatically.

An exploration of how a robot should model its human partner should begin by considering what information will be collected in this model. Our partner model contains three types of information: 1) a set of partner features; 2) an action model; and 3) a utility function. Partner features are used for partner recognition. This set of features allows the

robot to recognize the partner in subsequent interactions. The partner's action model contains a list of actions available to that individual. Finally, the partner's utility function includes information about the outcomes obtained by the partner when the robot and the partner select a pair of actions. Likewise, the self model also contains an action model and a utility function. The action model contains a list of actions available to the robot. Similarly the robot's utility function includes information about the robot's outcomes. The information encompassed within our partner models does not represent the final word on what types of information that should be included in such models. In fact, information about the partner's beliefs, knowledge, mood, personality, etc. could conceivably be included in these models. We use the dot (.) to denote the sets and functions within a model. Hence, $m^i.A^i$ denotes an action model contained within a partner model for individual i (see Figure 4.2 for an example).

Partner models contain information relating to Theory of Mind (Scassellati, 2002). Theory of mind describes the ability of an individual to attribute particular mental states to other individuals. The creation and maintenance of a partner model requires the ability to determine the reward and cost values for another individual as well as the actions available to the individual in a particular social situation. Thus, the creation of a partner model and the use of the partner model to populate an outcome matrix highlight the role Theory of Mind plays during social interaction.

The preceding discussion raises an important question: how do we measure partner model accuracy? For example, given a human partner with action set $m^{-i}.A^{-i}$ and utility function $m^{-i}.u^{-i}$, how close is the robot's partner model m^{-i} to the actual model $*m^{-i}$ where the * symbol is used to represent a target model. We address this problem by

viewing action models and utility functions as sets. The action model is a set of actions, $a_j^i \in A^i$, and a utility function, u^i , is a set of triplets, $(a_j^i, a_k^{-i}, \mathfrak{R})$, that contains the action of each individual and a utility value. We can then do set comparisons to determine the accuracy of the robot's partner model m^{-i} . Figure 4.3 presents a concrete example.

Measuring Partner Model Accuracy

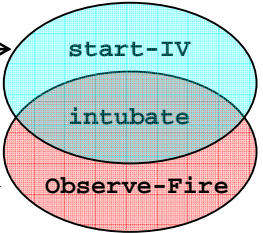
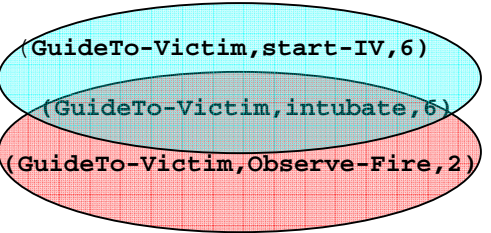
Models	Action Model Calculation	Utility Function Calculation
$*m^{-i}$ m^{-i}		
Operations 1. Cardinality 2. Cardinality 3. Set difference 4. Set intersect 5. Card. difference 6. Card. intersect	$ m^{-i} = 2$ $ *m^{-i} = 2$ $m^{-i} - *m^{-i} = \text{Observe-Fire}$ $*m^{-i} \cap m^{-i} = \text{intubate}$ $ m^{-i} - *m^{-i} = 1$ $ *m^{-i} \cap m^{-i} = 1$	$ m^{-i} = 2$ $ *m^{-i} = 2$ $m^{-i} - *m^{-i} = (\text{GuideTo-Victim, Observe-Fire, 2})$ $*m^{-i} \cap m^{-i} = (\text{GuideTo-Victim, intubate, 6})$ $ m^{-i} - *m^{-i} = 1$ $ *m^{-i} \cap m^{-i} = 1$
Calculation 7. Add values to the equation (1)	$d = \frac{\left(\frac{ m^{-i} - *m^{-i} }{ m^{-i} }\right) + \left(1 - \frac{ *m^{-i} \cap m^{-i} }{ m^{-i} }\right)}{2}$ $d^a = \frac{\left(\frac{1}{2}\right) + \left(1 - \frac{1}{2}\right)}{2} = \frac{1}{2}$	$d = \frac{\left(\frac{ m^{-i} - *m^{-i} }{ m^{-i} }\right) + \left(1 - \frac{ *m^{-i} \cap m^{-i} }{ m^{-i} }\right)}{2}$ $d^u = \frac{\left(\frac{1}{2}\right) + \left(1 - \frac{1}{2}\right)}{2} = \frac{1}{2}$
Overall 8. Calculate the overall model accuracy	$d^{-i} = \frac{d^a + d^u}{2} = \frac{\left(\frac{1}{2}\right) + \left(\frac{1}{2}\right)}{2} = \frac{1}{2}$	

Figure 4.3 The figure above provides an example measurement of partner model accuracy. The robot currently has an action model for partner consisting of two actions, one of which is not correct. The same is true of the robot's utility function for the partner. Calculations are provided in the lower half of the diagram. The resulting distance from the true model is $d=0.5$. A distance of zero means that the models are the same. A distance of one means that the model is completely dissimilar to the target model.

Two types of error are possible. Type I error (false positive) occurs if an action or utility is added to the robot's partner model (m^{-i}) which is not in the actual model ($*m^{-i}$). Consider as a running example interaction with an emergency medical technician (EMT) in a search and rescue situation. A type I error occurs when the robot believes that the EMT can do some action, such as observing a fire (Figure 4.3), which, perhaps because of the situation, it cannot. Type II error (false negative) occurs if an action or utility in the actual model ($*m^{-i}$) is not included in robot's partner model (m^{-i}). A type II error occurs when the robot does not know that the EMT can perform an action, such as starting an IV (Figure 4.3). Both types of error must be included in a measure of action model or utility function accuracy. Moreover, a utility function value was not considered present in the actual model if the value differed from the actual value by an arbitrary amount (we chose a value of one).

To determine Type I error we calculate the number of actions or utilities in m^{-i} which are not in $*m^{-i}$ as a percent of m^{-i} . Thus, $\frac{|m^{-i} - *m^{-i}|}{|m^{-i}|}$, is the number of actions in the robot's model that are not in the actual model divided by the number of actions in the robot's model.

Type II error can be calculated as the number of actions or utilities in both m^{-i} and $*m^{-i}$ as a percent of $*m^{-i}$. Thus $\frac{|*m^{-i} \cap m^{-i}|}{|*m^{-i}|}$ is the number of actions in both models divided by the number of actions in the actual model. As the number of actions in both

models increases, the accuracy increases. Hence, since we seek an measure of distance

(inaccuracy), the term $1 - \frac{|m^{-i} \cap m^{-i}|}{|m^{-i}|}$ is used.

Finally, the two types of errors are averaged in the equation,

$$d = \frac{\left(\frac{|m^{-i} - m^{-i}|}{|m^{-i}|} \right) + \left(1 - \frac{|m^{-i} \cap m^{-i}|}{|m^{-i}|} \right)}{2} \quad (1)$$

to create d , an overall measure of model accuracy and distance from the true model for either an action model (d^a) or a utility function (d^u). To determine overall model accuracy we average the error from both components of the partner model,

$$d^{-i} = \frac{d^a + d^u}{2}. \quad (2)$$

The value of d represents the distance of the robot's partner model from the actual model. When d equals zero the robot's partner model is equal to the true partner model. When d is equal to one then the robot's partner model is completely dissimilar (the intersection is empty) to the true partner model. To calculate the model accuracy we follow the steps in Figure 4.3. The first six steps from Figure 4.3 perform a series of set operations on the action model and utility function. The seventh step inserts the values obtained by the set operations into equation (1). In the final step, equation (2) is used to combine errors from both models.

Equation (2) weighs action model and utility function accuracy equally. We could have chosen to weigh the accuracy of either the action model or the utility function as more important in deciding overall partner model accuracy. As shown in the next chapter,

action model accuracy and utility function accuracy effect action selection differently. In the end, we chose to weigh action model accuracy equally to utility accuracy.

This section has introduced partner models and a method for measuring the difference between models. Partner models are critical when creating outcome matrices. The following section details a process by which outcome matrices are used to select actions.

4.3 The Transformation Process

Interdependence theory is based on the claim that people adjust their interactive behavior in response to their perception of a social situation's pattern of rewards and costs. Kelley noted that individuals often transform their interactions to include irrational aspects of socialization such as emotion and their internal predilections or dispositions (Kelley, 1979). Moreover, these internal transformations govern socialization and ensure that people are not simply rational outcome maximizers. Section 2.2.1 described a general architecture for social deliberation designed by interdependence theorists (Kelley & Thibaut, 1978; Rusbult & Van Lange, 2003). In this section we flesh out the details of this design, creating an architecture for social deliberation suitable for implementation on a robot.

As discussed briefly in section 2.2.1, interdependence theory presents a process by which the given situation is first perceived by the individual and then cognitively transformed, creating an effective situation on which action is based (Figure 2.5). Recall that the given situation is a perceived instance of one type of social situation. The effective situation, on the other hand, represents outcomes that include many various aspects of the individual's own internal predilections, such as his or her disposition. Behaviors are directly selected from the resulting effective situation. Between the given

situation and the effective situation a transformation process exists that alters the given situation to create the effective situation.

In this section, we detail a method for transforming an outcome matrix into a matrix that includes the robot’s own internal disposition. Disposition is defined here as a stable, social character manifested in an individual (section 7.1 describes disposition in more detail).

Framework for Social Action Selection

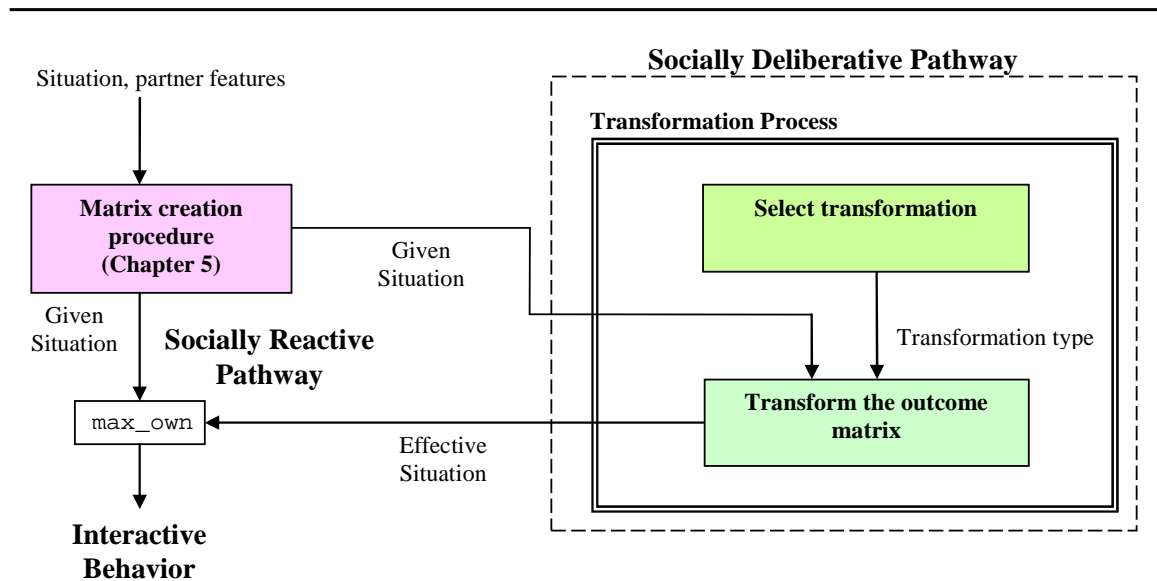


Figure 4.4 The figure above depicts our framework for social action selection. Situation features are used to generate the given situation (described in section 5.3). The given situation is transformed to include the robot’s disposition producing the effective situation. Finally, an action is selected from the effective situation.

Figure 4.4 depicts a framework for social action selection. This framework is similar to the general process delineated by interdependence theory (Figure 2.5). The process begins with the given situation. Our method for creating the given situation will be described in chapter 5. The given situation is represented with an outcome matrix that reflects a situation at a ‘gut’ or perceptual level without internal transforms. Once the individual has generated an outcome matrix representing the given situation this matrix can be used in one of two pathways. The socially reactive pathway is a developmental

pathway some researchers claim reflects the lack of social deliberation by children (Rusbult & Van Lange, 2003). In this pathway, the individual acts egotistically simply selecting the action that maximizes the individual's own outcome without consideration of the other individual. The second pathway we term the socially deliberative pathway. This pathway transforms the given situation into an effective situation on which the individual will act. This dissertation will focus on the second pathway. Before describing the remainder of the diagram, we will first describe how outcome matrices are transformed.

4.3.1 Transforming an outcome matrix

An outcome matrix, as a representation, affords many simple strategies for selecting an action from the matrix. The simplest of all strategies is to select the action that results in the greatest potential outcome for oneself. We term this strategy *max_own* because it serves to maximize the deciding individual's outcome without consideration of the partner.

Table 4.1 lists several other simple strategies. Each of the strategies listed in this table uses a simple computational process to alter the original matrix into a new matrix. For example, the *max_joint* transformation can be computationally described as, ${}_{xy}\hat{o}^1 = {}_{xy}o^1 + {}_{xy}o^2$ where x, y are constant and \hat{o} represents the transformed outcome value. This transformation replaces outcomes with the sum of the robot's and its partner's outcome. Moreover, each transformation has a particular social character that reflects the individual that chooses the transformation. For instance, an individual that often chooses to maximize its partner's outcome (*max_other*) is typically considered to be acting

altruistically. In other words, the continued selection of a single transformation reflects an individual's disposition.

Table 4.1 The table provides a list of some example transformations. The table provides the name of the transformation, a description, the computational arrangement of the transformation, and the character displayed by an individual that often selects the type of transformation. A more complete list is provided in Appendix C.

Example Transformations

Transformation name	Transformation description	Computational mechanism	Social character
<i>max_own</i>	No change	No change	Egoism —the individual selects the action that most favors their own outcomes
<i>max_other</i>	Swap partner's outcomes with one's own	${}_{xy}\hat{o}^1 = {}_{xy}o^2$	Altruism —the individual selects the action that most favors their partner
<i>max_joint</i>	Replace outcomes with the sum of the individual and the partner's outcome	${}_{xy}\hat{o}^1 = {}_{xy}o^1 + {}_{xy}o^2$	Cooperation —the individual selects the action that most favors both their own and their partner's outcome
<i>min_diff</i>	Maximize the value of the action that has the minimal difference to that of the partner.	${}_{xy}\hat{o}^1 = \max\left(\left {}_{xy}o^1 - {}_{xy}o^2\right \right)$ $- \left {}_{xy}o^1 - {}_{xy}o^2\right $	Fairness —the individual selects the action that results in the least disparity

Returning to our framework, the outcome matrix representing the given situation is transformed using one of the transformations listed in Appendix C to create the effective situation. This occurs as a two-step process. The first step is to select a transformation. Once a transformation is selected the outcome matrix is transformed according to the rules of the transformation. Formally,

$$O_E = f(O_G, \theta), \quad (3)$$

where is O_E the effective outcome matrix, O_G is the given outcome matrix, θ is the transformation, and the function f transforms the matrix. Once the matrix has been

transformed, the action that maximizes the individual's outcome is selected, completing the social deliberative pathway.

Consider, for example, a robot and a human foraging for two types of objects in a household environment. The human searches for only blue objects, while the robot searches for either blue or red objects but prefers red objects. The given situation for the robot is created by counting the number of colored objects in each room. If the robot uses the *max_own* strategy then it will select the room with the bluest objects. We can, however, transform the matrix to make the actions of the robot more helpful for the human. If we use a *max_joint* transformation, then the robot will select the room with the most red and blue objects to forage in. If, on the other hand, we use a *max_other* then the robot will select the room with the reddest objects to forage in.

As will be shown, this transformation process is a simple yet powerful way for a robot to not only reason about its own social actions but to also reason about the actions of its human partner.

This chapter has presented the conceptual underpinnings for our framework for human robot social interaction. We have presented a method of representing interactions computationally, for modeling the robot's interactive partner, and for selecting social actions. The chapter offers the groundwork for addressing this dissertation's research questions. The chapters that follow will build on this framework, presenting methods for using these concepts and results showing that the concepts work.

CHAPTER 5

FROM PERCEPTION TO OUTCOME MATRIX

As a theoretical tool a representation may be of value simply because it lends insight into the computational problem itself (Hutchins, 1995). Yet, to be useful to the field of robotics, it must also be possible to create instances of the representation from the noisy, uncertain perceptual input available to a robot. Because the challenge of creating outcome matrices from social interaction in general is vast, this chapter does not mark the final word on the subject. Rather, the chapter presents preliminary algorithms and insight into this problem and only attempts to show that it is possible to create our representation of social interaction from perceptual information. The bulk of this dissertation will then focus on using the outcome matrix to characterize trust and social relationships.

This chapter presents a series of algorithms for creating outcome matrices from perceptual information. The first algorithm assumes accurate knowledge of the partner and the environment, but serves as a profitable place to begin developing a general purpose algorithm for generating outcome matrices. We use this algorithm to explore the sensitivity of the outcome matrix to different types of error. The next two algorithms make fewer assumptions and demonstrate that it is possible to create outcome matrices. The chapter concludes with a discussion of the open problems and future challenges related to the creation of outcome matrices.

5.1 Developing an Algorithm for Outcome Matrix Creation

Experts note that a representation of knowledge acts as a surrogate for a naturally occurring phenomena (Davis, Shrobe, & Szolovits, 1993). As a surrogate, a representation maintains specific types of information about the phenomena and omits

other types of information. When developing an algorithm for the creation of a representation, it is therefore natural to ask what types of information are present in the representation.

Recall that an outcome matrix is our representation for social interaction. The previous chapters detailed our reasons for choosing the outcome matrix as a representation for social interaction. The information represented in an outcome matrix centers on three questions: 1) who is interacting? 2) What actions are available to each individual? And 3) how will the selection of a pair of actions influence each individual? Moreover, these questions must be answered in order because, for example, the identity of the robot's partner could influence which actions are available to a partner.

General Matrix Creation Algorithm

Input: Self model m^i and partner model m^{-i} .

Output: Outcome matrix O .

1. Create empty outcome matrix O
2. **Set** $O.partner = g(m^{-i}.features)$ //Use perceptual features to retrieve partner name
3. **Set** $O.self = \text{"robot"}$ //Assign robot as name of self
4. **Set** $O.columns = m^i.A^i$ //Use model of self to set actions for self
5. **Set** $O.rows = m^{-i}.A^{-i}$ //Use model of partner to set actions for self
6. **For** each action pair (a_j^i, a_k^{-i}) in A^i, A^{-i}
7. $O^i(a_j^i, a_k^{-i}) \leftarrow m^i.u^i(a_j^i, a_k^{-i})$ //Use utility function to assign outcome values
8. $O^{-i}(a_j^i, a_k^{-i}) \leftarrow m^{-i}.u^{-i}(a_j^i, a_k^{-i})$ //Use partner utility function to assign partner's
9. **End** //outcome values
10. **Return** O

Box 5.1 The algorithm above creates an outcome matrix from the input partner and self models. The algorithm operates by successively filling in the elements of the matrix. The function x is a mapping from partner features to ID.

We have thus sketched the outline of an algorithm for creating outcome matrices. Box 5.1 depicts the general form of the algorithm. The algorithm takes as input the self model and the partner model and produces an outcome matrix as output. The self model is a mental model the robot maintains of itself. It contains the robot's action model and a

utility function. The action model consists of a list of actions available to the robot. Similarly the robot's utility function includes information about the robot's outcomes. The first step of the algorithm creates an empty outcome matrix. Next the algorithm sets the partner's ID and both the robot's and the partner's actions. This step uses the function g to map perceptual features to a unique label or ID. ID creation provides a means of attaching the perception of an individual to what is learned from interacting with that individual. In theory, any method that provides a unique ID from perceptual features should work in this algorithm. We have not, however, explored this claim experimentally. Finally, for each pair of actions in the action models, we use each individual's utility function (u^i and u^{-i}) to assign an outcome for the pair of actions.

Consider, as a running example, a firefighter in a search and rescue situation. The firefighter's action model contains action for performing CPR, fighting fires, rescuing people, etc. The firefighter's utility function indicates the he ranks actions which save people (such as performing CPR) as more valuable than actions which reduce property damage (such as fighting a fire). Perceptual features such as having an axe, an oxygen tank, and a helmet, indicate that the person is a firefighter. Other features such as height and hair color identify the firefighter as a particular individual. Figure 5.1 demonstrates the use of the Outcome Matrix Creation algorithm for creating an outcome matrix for the firefighter and an assisting robot in a search and rescue environment.

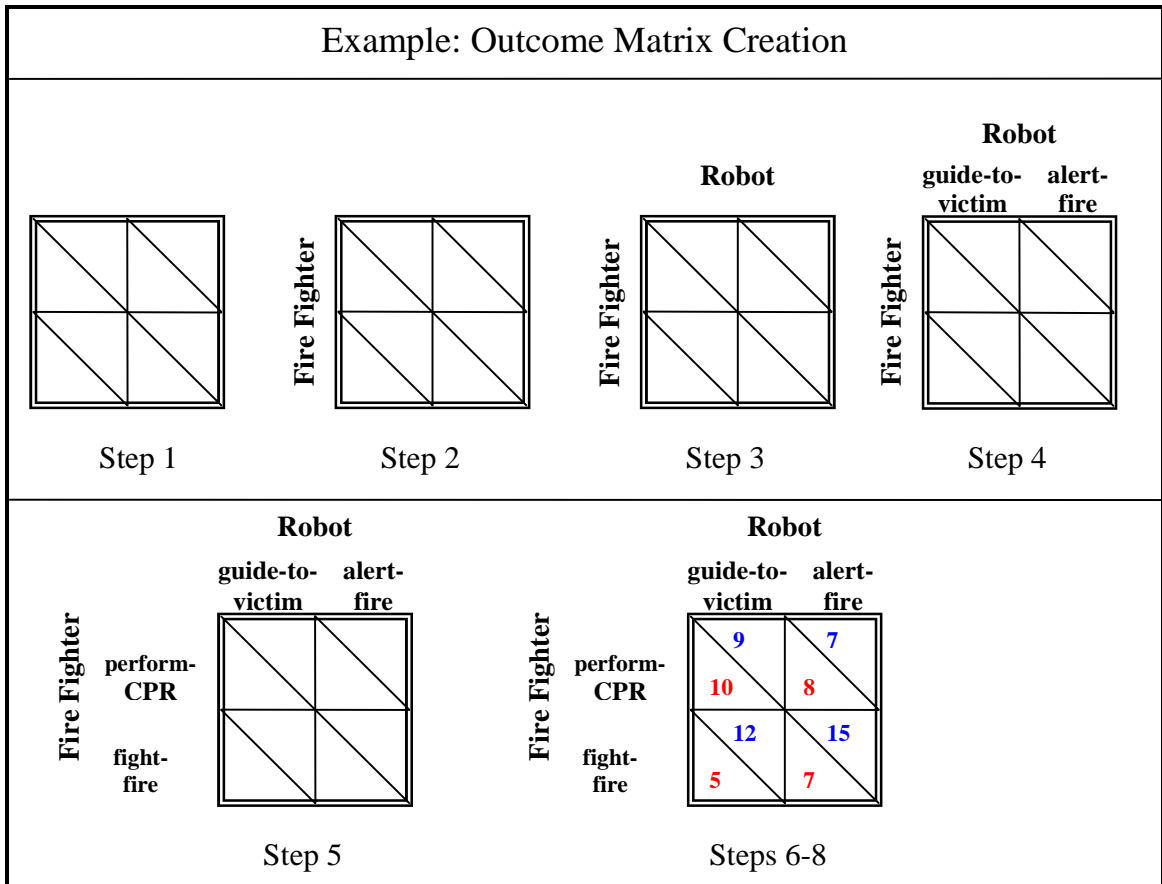


Figure 5.1 An example of the Outcome Matrix Creation algorithm in a search and rescue environment with a firefighter. Step 1 begins with an empty matrix which is filled with information related to the interaction. The result is the final matrix labeled Steps 6-8.

It should be apparent that the Outcome Matrix Creation algorithm simply fills in the matrix with missing information. Moreover, the accuracy of the outcome matrices created by the algorithm depends entirely on the accuracy of the information contained in the self and partner models. The question of creating outcome matrices then becomes a question of how do we create accurate partner models. Stated another way, if accurate partner models could be created then we could use the Outcome Matrix Creation algorithm to create accurate outcome matrices. For this reason, sections 5.3 and 5.4 present algorithms for creating and refining partner models. Before exploring these algorithms, however, we will use the Outcome Matrix Creation algorithm to examine the sensitivity

of the outcome matrix as a representation of social interaction to different types of error. This will, in turn, influence the design of our later algorithms.

5.2 Outcome Matrix Error Sensitivity

Chapters 2 and 4 discussed the outcome matrix from a historical perspective and from the perspective of related work. In these chapters we argued that the outcome matrix is a feasible representation of interaction because it has been used as such in other disciplines for decades. In this section we present empirical results supporting our assertion that the outcome matrix is an excellent representation for interaction on a robot. These results will focus on the representation's sensitivity to different types of error. We define sensitivity here with respect to action selection because it is a robot's ability to select and perform actions that will likely have the largest repercussions on its human partner. Embodiment and reliance on potentially noisy sensors makes an examination of a representation's sensitivity to error an important consideration if the representation is to be used on a robot for two reasons. First, if the outcome matrix representation is sensitive to several types of errors then perhaps the representation is not suitable for implementation on robots. Second, different types of errors could affect the usefulness of outcome matrix in different ways and thus impact our outcome matrix generation algorithm.

The purpose of this section is to examine how the introduction of errors into the outcome matrix impacts a simulated robot's ability to select actions. Ideally, for every error introduced into our representation, less than one action selection error will occur. We use the term error to denote any difference in the information contained within a representation from the target model. Hence, errors can include incorrect values or missing information. We consider the outcome matrix to be sensitive to a specific type of error if the action selection error increases linearly or greater with respect to the error introduced.

We conducted simulation experiments to explore the sensitivity of the outcome matrix to different types of error. We explored four different types of errors: errors in outcome value magnitude, errors in single outcome values, action insertion errors, and action deletion errors. Errors in outcome value magnitude occur when all of the outcome values for a partner are multiplied by some value $k \in \mathfrak{R}$. For example, if all of the partner’s outcome values are changed to be half of the true value. Errors in single outcome values occur when one or many particular outcome values differ from their true value. This error occurs, for example, when the robot incorrectly values a particular action human-robot action pair. Action insertion errors occur if the partner’s action model includes actions which could not be used in the current social situation. Using the firefighter example from the previous section, if the robot believes that the firefighter is capable of performing actions such as administering an IV when, in fact, the firefighter cannot. Action deletion errors, on the other hand, occur when actions that could have been used in the current social situation are deleted from the partner’s current action model. For example, not recognizing that a firefighter can fight fires would result from an action deletion error.

Table 5.1 Environment types, partner types, and robot types for the outcome matrix error sensitivity experiment.

Environment Type	Partner Type	Robot Type
assistive	police officer	police officer assistant
household	firefighter	firefighter assistant
museum	accident victim	medical assistant
prison	hospital patient	
search and rescue	citizen	
	medical staff	

The same general experimental procedure was used to investigate all four different types of error in four different experiments. The USARSim simulation environment, robot models, and tools described in section 3.4.1 were used. To ensure generality with respect to the type of environment, the experiments were conducted in all five simulation

environments (Table 5.1). Similarly, to ensure generality with respect to the partner and self models, we created three different types of robots and six different types of human partners (Table 5.1). Each partner and self model had a different set of actions capable of being performed in a particular environment. For example, in the search and rescue environment the police assistant robot produced an auditory alarm if it found a victim, whereas in the household environment the police assistant robot searched the household for burglars. Overall 90 ($3 \times 6 \times 5$) different combinations of robot type (3), partner type (6), and environment (5) were created. Each unique combination affected the action model and utility function for the robot and its partner. These 90 models served as target models for each of the four experiments.

Table 5.2 Example actions for different types of individuals.

Partner Type	Example actions	Robot type	Example actions
police officer	perform-CPR, arrest-person, search-home	police assistant	alert-guards, alert-security, observe-exhibit
firefighter	perform-CPR, fight-fire, rescue-person	firefighter assistant	guide-to-fire, guide-to-victim
accident victim	crawl, limp, moan	medical assistant	guide-to-victim, guide-to-triage, light-victim, light-triage
hospital patient	get-food, do-art-therapy, watch-TV		
citizen	watch-scene, talk, run-away		
medical staff	stabilize-person, treat-illness, assess-person		

The USARSim model of the Pioneer DX robot was used in all experiments (Figure 3.4). The robot had both a camera and a laser range finder. The medical assistant robot type had a light for communicating the location of victims, the police assistant robot type had an auditory alarm, and the fire assistant robot type had neither a light nor an alarm. Feedback from the simulation environment provided localization information. The robot used speech synthesis to communicate questions and information to the human partner. Speech recognition translated the spoken information provided by the human. Microsoft’s Speech SDK provided the speech synthesis and recognition capabilities.

Table 5.2 lists example actions available to each type of robot. Note that the suitability of an action depends on the type of environment. The mapping from action to environment was created by reasoning about the types of actions that could be performed by this robot in a particular environment.

The robot's human partner used the interface depicted in Figure 3.2 to interact with the robot. This interface was developed from an existing USARSim tool (Zaratti, Fratarcangeli, & Iocchi, 2006). The interface allows the human to move around and view the environment. The human interacted with the robot by speaking a predefined list of commands. Table 5.2 lists example actions available to the human. The action set for the human was derived by reasoning about the types of actions that would be available to a police officer, firefighter, victim, citizen, medical staff, and hospital patient in each of the environments.

Utility functions for both the human and the robot were created by producing an arbitrary ordering of the individual's actions in an environment in a list format. Next, each action in the list received a utility equal to its position in the list (beginning at zero). Finally, a value equal to the half of the size of the list was subtracted from each of the utilities on the list. The purpose of subtracting this value was to ensure that roughly half of the actions had negative utility. For example, if the robot's action list was (guide-to-victim, guide-to-triage, light-victim, light-triage) the resulting utilities for each action would be (-2,-1,0,1). Action pairs received a utility equal to the sum of utility for each individual action.

Like the robot, the human's utility function was created by producing an arbitrary ordering of the actions in an environment and setting the action in the middle of the ordering to zero utility.

The following general procedure was used for each of the four different experiments:

Table 5.3 Experimental procedure for the outcome matrix error sensitivity experiments.

Experimental Procedure	
1)	Create 90 target models reflecting the different combinations of robot type, partner type, and environment.
2)	For each target model create noisy models by injecting the model with random Gaussian error. The amount of error introduced to the model is the independent variable. The type of error depended on the experiment.
3)	The robot gathers information about its type, the environment, and the partner. Specifically, the human operator informs the robot of its robot type (police assistant, firefighter assistant, medical assistant), the robot uses OpenCV to detect the presence or absence of objects to determine the type of environment, and the robot uses speech synthesis and recognition to query the partner for their type.
4)	Having obtained the robot type, partner type and environment type, the robot retrieves from memory a (possibly error laden) partner and a self model.
5)	The General Matrix Creation algorithm (Box 5.1) is used to create an outcome matrix.
6)	The robot uses a <i>max_own</i> action selection strategy (see section 4.3.1 for more details on action selection strategies) selecting the action that maximizes its own outcome without regard to the partner.
7)	The action is performed in the environment. The robot's action selection and resulting outcome values are recorded. The robot queries the human for his/her action selection and records the response.
8)	Steps 3-7 are repeated for each of the 90 combinations of robot type, partner type, and environment.
9)	After each combination of robot type, partner type, and environment has been tested, the error introduced into the models is increased by 5 percent and steps 3-8 are repeated.
10)	Continue until the model is 90 percent error.

In each of the experiments, the percentage of error introduced to the robot's target models is the independent variable. The percentage of incorrect actions selected by the robot is

the dependent variable. An incorrect action is an action which differs from the action that would have been selected had the error free target model been used instead of the error-laden model. The hypothesis tested for each experiment depended on the type of error investigated. In all experiments, analysis was performed by comparing the action selection error rate averaged over all environments to a linearly increasing rate. Figure 5.2 summarizes the model creation process for all four experiments.

Partner model creation procedure in the Outcome Matrix Error Sensitivity Experiments

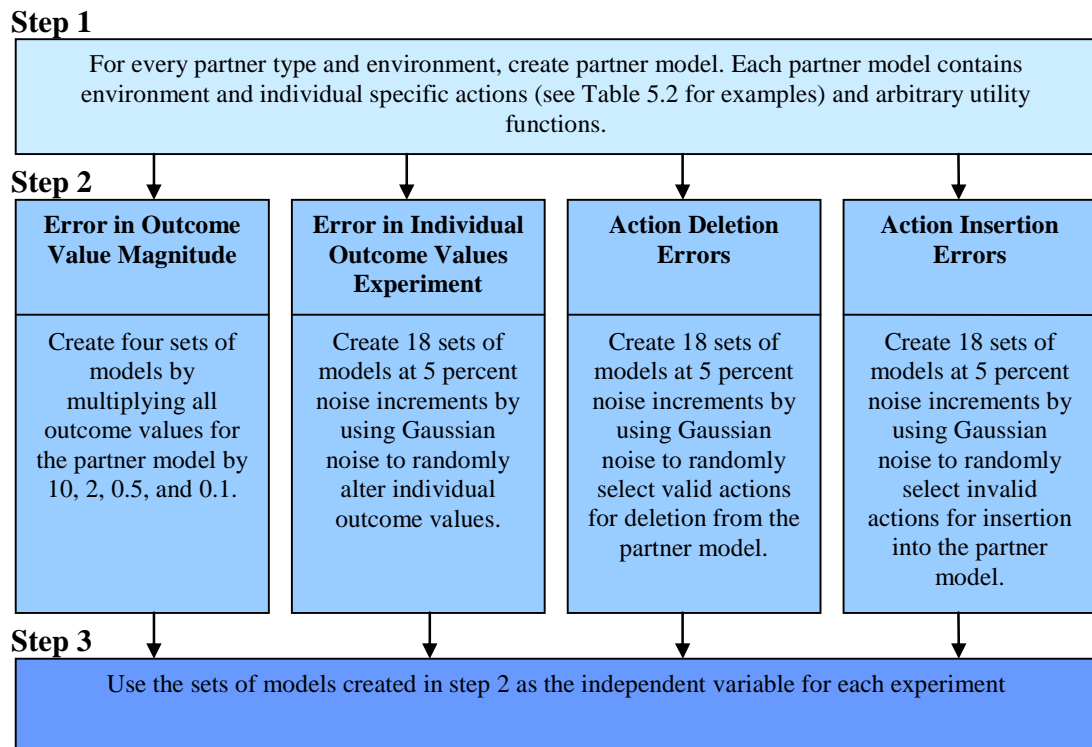


Figure 5.2 Diagram depicting the process used to create the partner models for the four error sensitivity experiments presented in the following four subsections.

5.2.1 Errors in outcome value magnitude

A common concern about the outcome matrix as a representation for interaction is that the outcome values are likely to be inaccurate. The following experiment explores this

concern by investigating the action selection resulting from errors in outcome value magnitude. Table 5.4 provides a summary of the experiment.

Errors in outcome value magnitude occur when the robot either uniformly inflates or deflates the rewards or costs associated with all action pairs within a matrix. These types of errors are common in human psychology (Sears, Peplau, & Taylor, 1991).

As mentioned above, an error of magnitude occurs when all of the outcome values for a model are altered by some value $k \in \mathfrak{R}$. Consider the example matrix from Figure 5.3. Using a *max_own* action selection strategy the robot would select the `alert-fire` action because the action pair (`alert-fire`, `fight-fire`) results in the largest outcome for the robot. Even if a utility function error results in an increase of all outcomes by a factor of 10, the robot will still select the same action. Moreover, the same is true of many types of systematic error (dividing by a positive value, multiplying by a positive number, etc.) that alters the magnitude of all values but does not alter their overall rank order in terms of outcome. We hypothesized that these types of errors would not affect the ability of the robot to select the correct action.

Table 5.4 Experiment summary for the errors in outcome value magnitude experiment.

Experiment Summary Errors in Outcome Value Magnitude	
Purpose	Investigate outcome matrix sensitivity to increases and decreases in magnitude of all outcome values.
Experiment Type	USARSim simulation
Hypothesis	Percentage of action selection errors is zero regardless of the magnitude of outcome value errors
Procedure	<ol style="list-style-type: none"> 1) Figure 5.2, Errors in Outcome Value Magnitude, used to create error models. 2) Procedure from Table 5.3 was used to perform experiment.
Independent variable	Magnitude of outcome value change. Multiplication factors were 10, 2, 0.5, and 0.1.
Dependent variable	Percentage of incorrect actions selected
Method of Analysis	Target model comparison
Conclusion	Hypothesis is supported. Errors are zero regardless of magnitude of error introduced.

Example Interaction

		Robot	
		guide-to- victim	alert- fire
Human Partner	perform- CPR	9 4	7 6
	fight- fire	12 5	15 7

Figure 5.3 An example outcome matrix from the error sensitivity experiments. The robot’s use of a *max_own* strategy would result in selection of the *alert-fire* action.

To test this hypothesis, we used the procedure from Figure 5.2 to create models with errors of four different magnitudes: 10, 2, 0.5, and 0.1. Next the procedure from Table 5.3 was used to perform experiment. The independent variable in this experiment was the magnitude of error, either 10, 2, 0.5, and 0.1. The dependent variable was the percentage of incorrect actions selected. We found the error rate to be zero regardless of the magnitude of change. This result is not surprising and simply reflects the fact that utility values form a preference relation with respect to the action possibilities being decided. We can conclude that outcome matrices are not sensitive to errors which impact the magnitude of all outcome values equally.

5.2.2 Errors in individual outcome values

Errors in magnitude affect all outcome values. What about errors that do not affect all outcome values? In contrast to an error in magnitude, errors with respect to individual outcome values may result in a new preference relation over actions. These errors occur when the robot’s utility function generates inaccurate values with respect to particular action pairs. Because of the noise associated robotic perception, action pair valuation uncertainty, and a lack of knowledge related to the partner, this type of error is expected

to commonly occur and therefore demands examination. Table 5.5 provides a summary of the experiment.

Table 5.5 Experiment summary for the errors in individual outcome values experiment.

Experiment Summary Errors in Individual Outcome Values	
Purpose	Investigate outcome matrix sensitivity to errors in individual outcome values.
Experiment Type	USARSim simulation
Hypothesis	The number of action selection errors is less than one per error in outcome value.
Procedure	<ol style="list-style-type: none"> 1) Figure 5.2, Errors in individual Outcome Value, used to create error models. 2) Procedure from Table 5.3 was used to perform experiment.
Independent variable	Percentage of outcome values replaced with error value.
Dependent variable	Percentage of incorrect actions selected.
Method of Analysis	Target model comparison
Conclusion	Hypothesis is supported. The rate of action selection errors per outcome values replaced is less than one.

Coming back to our firefighter example in Figure 5.3, the robot's outcome value for the action pair (alert-fire, fight-fire) is 15. For our purposes an error occurs whenever the robot believes the value for this action pair to be less than or equal to 14 or greater or equal to 16. Also notice that the action pair (alert-fire, fight-fire) results in the greatest potential outcome for the robot. A robot using the *max_own* strategy would thus select the alert-fire action. For this example matrix, an action selection error only occurs if the robot selects the guide-to-victim action. A robot using the *max_own* strategy would only select the guide-to-victim action under one of two conditions: 1) the outcome value of the action pair (alert-fire, fight-fire)=15 was less than the value of the action pair (guide-to-victim, fight-fire)=12 or 2) the outcome value of either the action pair (guide-to-victim, perform-CPR)=9 or (guide-to-victim, fight-fire)=12 was greater than the outcome value of (alert-fire, fight-fire)=15. Notice that of the possible perturbations of the outcome value

for the action pair (alert-fire, fight-fire) few would result in action selection errors. We therefore hypothesized that less than one action selection error results from an error in outcome value.

To test this hypothesis, we used the procedure from Figure 5.2 for the Errors in Individual Outcome Values Experiment to create partner models with different amounts of error added. Error was added by first using a Gaussian distribution to randomly select an outcome value within the partner model's utility function. Next, a Gaussian distribution was used to select a random value within the range of $[-20,20]$. If the new value differed from the original value, then the original value was replaced, creating an error. The process was continued until the desired amount of error had been introduced into the utility function. We created 18 sets of partner models with 5 percent increments of error added to the models ranging from 0 percent error to 90 percent error. Next the procedure from Table 5.3 was used to perform experiment. The independent variable in this experiment was the percent of error added. The dependent variable was the percentage of incorrect actions selected.

Figure 5.4 shows the results for this experiment. The bold black line in Figure 5.4 depicts the average result over all five environments. Thinner lines depict the results for individual environments. The bold white line provides a baseline by depicting an error rate of one error in action selection per error in outcome value. The experiment supports our hypothesis if the bold black line is below the bold white line. The results presented in Figure 5.4 indeed depict the bold black below the bold white line. The graph shows that the percent of error in outcome value increases from 0 to 90 percent, the rate of increase

in action selection error is less than linear (the bold white line). Thus, the experiment supports our hypothesis.

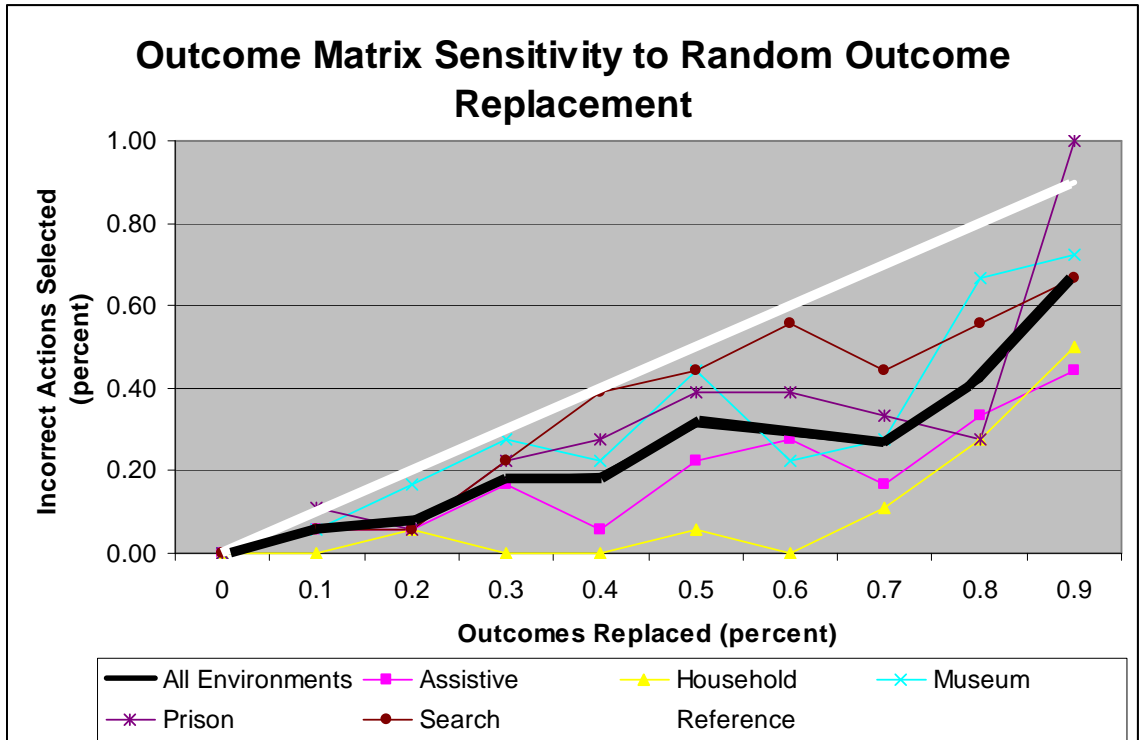


Figure 5.4 The graph depicts the percent of incorrect actions selected as a function of errors in outcome values. A y-axis value of 1.00 represents total selection of incorrect actions. The bold black line depicts the average incorrect actions selected for all environments. The individual colored lines represent changes in accuracy for each different environment. The bold white line is a baseline for comparison, depicting a linear decrease in accuracy. The fact that the bold black line is below the bold white line indicates that errors in outcome value result in less than linear action selection error.

In conclusion, this experiment demonstrates that outcome value inaccuracy has a less than linear effect on the robot’s ability to select actions. This is an important result. It indicates that our representation of interaction can be partially inaccurate (in terms of outcome values) and yet the robot will still select the correct action. To be more precise, even if we replace half of the outcome values within a matrix with incorrect values, the robot will still select the correct action 65% of the time. It also indicates that, in order to produce say 81% correct action selection, we should strive to have between 60-70% correct outcome values. Notice also that the bold black line in Figure 5.4 is not linear.

Rather, at 70% outcome value error the curve increases superlinearly, indicating a threshold after which the accuracy of the representation does not degrade gracefully. Overall, these results support our assertion that the outcome matrix is indeed a good representation for human-robot interaction.

5.2.3 Action deletion errors

We now consider errors related to the action model. In addition to errors involving outcome values, the action models from which the outcome matrix is constructed can be flawed. In this case, valid actions may have been left out or omitted from the matrix.

Table 5.6 provides a summary of the experiment.

Table 5.6 Experiment summary for the action deletion experiment.

Experiment Summary Action Deletion Errors	
Purpose	Investigate outcome matrix sensitivity to action model inaccuracy in the form of action omissions.
Experiment Type	USARSim simulation
Hypothesis	The number of action selection errors is approximately one per action deletion error.
Procedure	1) Figure 5.2, Action Deletion Errors, used to create error models. 2) Procedure from Table 5.3 was used to perform experiment.
Independent variable	Percentage of actions deleted.
Dependent variable	Percentage of incorrect actions selected.
Method of Analysis	Target model comparison
Conclusion	Hypothesis is true. The rate of action selection errors per actions deleted is approximately one.

An action deletion error occurs when an action, suitable for the robot's environment, has been left out of the matrix. This type of error can occur whenever the robot lacks a good model of its own actions. Even more likely, the matrix may contain omissions with respect to the actions of the robot's partner. The effect of action deletion errors with respect to the partner depends on the action selection strategy. The deletion of any one action only affects the matrix's accuracy when the action that would have otherwise been

selected is deleted. Returning again to the example in Figure 5.3, the robot chooses between two actions, `alert-fire` and `guide-to-victim`. If the robot is using a *max_own* strategy then omission of the `guide-to-victim` action is irrelevant as it would not have been chosen anyway. Only a deletion of the `alert-fire` action affects the robot's action selection accuracy. Thus, for example, if 10 percent of the robot's actions are deleted from the action model, then there is a 10 percent probability that the most favorable action has been deleted. The same is true if 30, 40, or 50 percent of the robot's actions are deleted from the action model. In each of these cases, the probability that the most favorable action has been deleted is equal to the percentage of actions deleted. We therefore hypothesis that the action selection error rate will be equal to the action deletion rate.

To test this hypothesis, we used the procedure from Figure 5.2 for the Action Deletion Errors Experiment to create partner models with differing action deletion error rates. Error was added by first using a Gaussian distribution to randomly select an action within the robot's action model. The selected action was then deleted. The process was continued until the desired amount of error had been introduced into the action model. We created 18 sets of self models with 5 percent increments of error added to the models ranging from 0 percent error to 90 percent error. Next the procedure from Table 5.3 was used to perform experiment. The independent variable in this experiment was the percent of error added. The dependent variable was the percentage of incorrect actions selected.

Figure 5.5 presents the results for this experiment. The bold black line in Figure 5.5 again indicates the average result over all five environments. Thinner lines portray the results for individual environments. The bold white line provides a baseline by depicting

an error rate of one error in action selection per error in action deletion. The experiment supports our hypothesis if the bold black line is approximately equal to the bold white line. The results presented in Figure 5.5 indeed confirm that bold black line is approximately equal to the bold white line. The graph shows that the percent of action deletion errors increasing from 0 to 90 percent, the rate of increase in action selection error is approximately linear (the bold white line). Thus, the experiment supports our hypothesis.

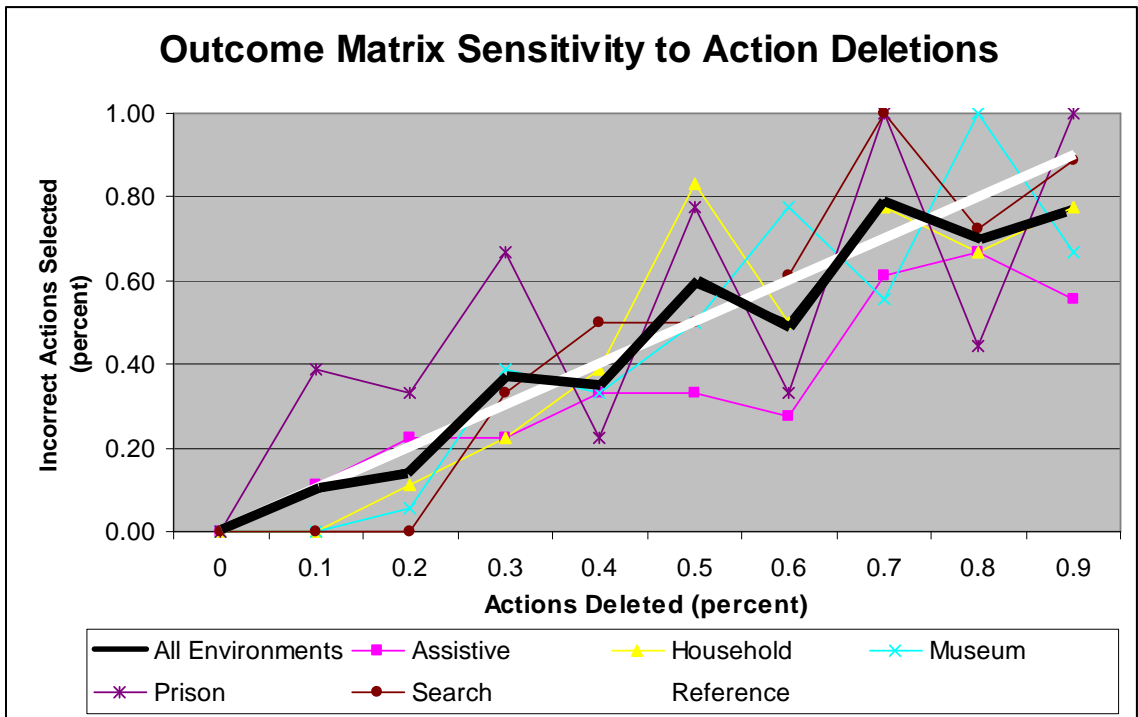


Figure 5.5 The graph depicts the percent of incorrect actions selected as a function of increasing random action deletion. The bold black line depicts the average incorrect actions selected for all environments. The individual colored lines represent changes in accuracy for each different environment. The bold white line is a baseline for comparison, depicting a linear decrease in accuracy. Note that the black line approximates the white line. Hence, in contrast to the two previous experiments this type of error increases approximately linearly.

In conclusion, this experiment demonstrates that action deletion inaccuracy has approximately a linear effect on the robot’s ability to select actions. Unlike the two previous experiments, each action deletion error results in approximately one action selection error. The impact of action deletion errors on action selection accuracy is thus

greater than the impact of either errors in magnitude or errors in individual outcome values. This fact will help to shape our creation of an algorithm for creating outcome matrices.

5.2.4 Action insertion errors

An outcome matrix can also contain actions that are not possible given the type of environment. Moreover, because each invalid action results in several invalid outcome values, these types of errors have the potential to flood the outcome matrix with improper outcome values. Table 5.7 provides a summary of the experiment.

Table 5.7 Experiment summary for the action insertion errors experiment.

Experiment Summary Action Insertion Errors	
Purpose	Investigate outcome matrix sensitivity to action model inaccuracy in the form of action insertion errors.
Experiment Type	USARSim simulation
Hypothesis	The number of action selection errors is less than one per action insertion error.
Procedure	1) Figure 5.2, Action Insertion Errors, used to create error models. 2) Procedure from Table 5.3 was used to perform experiment.
Independent variable	Percentage of invalid actions inserted into the action model.
Dependent variable	Percentage of incorrect actions selected.
Method of Analysis	Target model comparison
Conclusion	Hypothesis is true. The rate of action selection errors per actions inserted is less than one.

Assuming a *max_own* action selection strategy, an action insertion error results in the incorrect selection of an action only if the new action adds a new maximum value to the matrix. Referring back to Table 5.2, *watch-TV* is an action used by human hospital patients. For a robot in a search and rescue environment, an error occurs if the action is added to the robot's action model. This action insertion error would add another column to the matrix along with outcome values for each action pair. Because an incorrectly added action may not result in a new maximum value for the matrix, we hypothesized

that the percentage of incorrect actions selected by the robot would be less than the rate errors inserted.

To test this hypothesis, we used the procedure from Figure 5.2 for the Action Insertion Errors Experiment to create partner models with differing action insertion error rates. Error was added by first using a Gaussian distribution to randomly select an action from a global pool of actions used by both the robot and the human. If the action was not capable of being performed in the environment, then the selected action inserted into the robot's action model resulting in an error. The process was continued until the desired amount of error had been introduced into the action model. We created 18 sets of self models with 5 percent increments of error added to the models ranging from 0 percent error to 90 percent error. Next the procedure from Table 5.3 was used to perform experiment. The independent variable in this experiment was again the percent of error added. The dependent variable was the percentage of incorrect actions selected.

Figure 5.6 presents the results for this experiment. The bold black line in Figure 5.6 again indicates the average result over all five environments. The bold white line provides a baseline by depicting an error rate of one error in action selection per error in action insertion. The experiment supports our hypothesis if the bold black line is below the bold white line. The results presented in Figure 5.6 indeed confirm that the bold black line is below the bold white line. The graph shows that as the percent of actions incorrectly inserted into the outcome matrix increases from 0 to 90 percent of the total actions within the matrix, the rate of increase in action selection error is less than linear (the bold white line). Thus, the experiment supports our hypothesis.

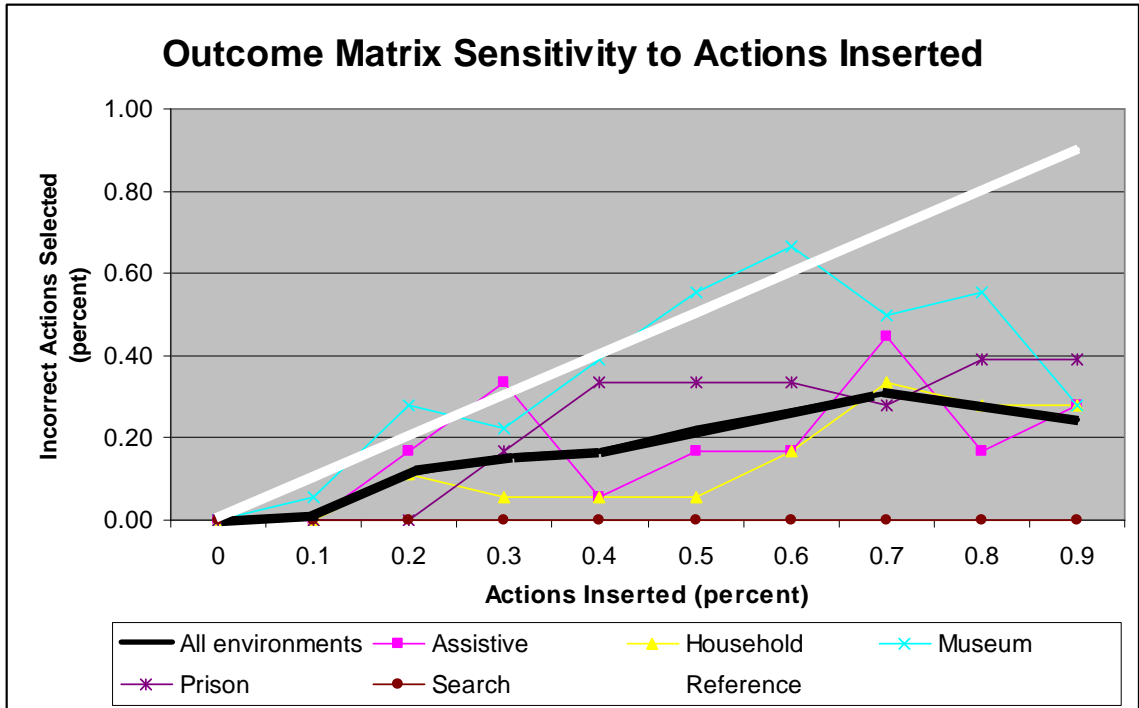


Figure 5.6 The graph depicts the percent of incorrect actions selected as a function of increasing random action insertion. As in the other graphs the bold black line depicts the average incorrect actions selected for all environments, the colored lines represent the changes in accuracy for each different environment, and the bold white line is a baseline for comparison, depicting a linear decrease in accuracy.

In conclusion, this experiment demonstrates that action insertion errors have a less than linear effects on the robot’s ability to select actions. To be more precise, only 22 percent action selection error results when 90 percent of actions within the robot’s action model are invalid. Thus, action deletion errors will result in greater action selection error than action insertion errors. These results further support our assertion that the outcome matrix is indeed a good representation for human-robot interaction.

5.2.5 Error sensitivity conclusions

This section has explored the sensitivity of the outcome matrix to different types of errors. Our purpose was to determine 1) if the outcome matrix representation is sensitive to several types of errors and 2) if different types of errors affect the usefulness of outcome matrix. The results are important in that they will impact the development our

algorithm for matrix creation which we present in the two remaining sections of this chapter. Moreover, the results demonstrate that as errors in outcome value magnitude, specific outcome values, and action insertion errors accumulate within an outcome matrix, action selection errors increase at a lesser rate. Had we instead found outcome matrices to be sensitive to these types of errors, because of the uncertainty inherent in a robot's perceptual abilities and in interaction, it might have been necessary to abandon this approach to human-robot interaction. We did, however, determine that outcome matrices are sensitive to action deletion errors. We will use this information to avoid action deletions when crafting our algorithm for outcome matrix creation.

5.3 The Interact and Update Algorithm

Section 5.1 presented the Outcome Matrix Creation algorithm as a means of creating outcome matrices from partner models. It should be apparent that the Outcome Matrix Creation algorithm simply fills in the matrix with missing information. Moreover, the accuracy of the outcome matrices created by the algorithm depends entirely on the accuracy of the information contained in the self and partner models. This begs the question, where does the information for the models come from? The interact-and-update algorithm serves this purpose.

The interact-and-update algorithm uses information learned during an interaction to revise its partner and robot models. Norman notes that humans continually revise their mental models with additional interaction (Norman, 1983). Our algorithm employs a similar strategy, updating its representation of its human partner with each additional interaction. The algorithm works by first predicting the action the partner will select and

the outcomes the robot and the partner will obtain. Then, in the update phase, the algorithm adjusts the partner model.

Interact-and-update Algorithm

Input: Partner features $f_1^{-i}, \dots, f_n^{-i}$ and situation features e_1, \dots, e_n

Pre-interaction

1. **Set** $m^i = y(e_1, \dots, e_n)$ //use situation features to retrieve self model
2. **Set** $m^{-i} = z(f_1^{-i}, \dots, f_n^{-i})$ //use partner features to retrieve model of partner
3. *OutcomeMatrixCreationAlgorithm*(m^i, m^{-i}) //from Box 5.1
4. **Set** $a^i = \max_own(O^i)$, $a^{-i} = \max_own(O^{-i})$ //set self and expected partner action
5. **Set** $o^{-i} = O^{-i}(a^i, a^{-i})$, $o^i = O^i(a^i, a^{-i})$ //set expected outcome

Interact

6. **Perform** a^i //perform action in the environment

Update

7. **Perceive** value a^{-i}, o^i, o^{-i} //Use perception to retrieve values for predicted values
8. **If** $a^{-i} \neq a^{-i}$ //If action is not what was predicted then
9. **update** $m^{-i}.A^{-i} = a^{-i}$, $m^{-i}.u(a^{-i}) = o^{-i}$ //update the model and utility function
10. **else if** $o^{-i} \neq o^{-i}$ //else if the outcome is not what was expected
11. **update** $m^{-i}.u(a^i, a^{-i}) = o^{-i}$ //update the utility function
12. **If** $o^i \neq o^i$ **then update** $m^i.u(a^{-i}, a^i) = o^i$
13. **for all** a^{-i} in m^{-i} //for each partner action in the partner model
14. **if** $p(a^{-i}) < k$ **then delete** a^{-i} //remove the action if the probability is suf. small

Box 5.2 Algorithm for using partner and self models to create outcome matrices. The algorithm successively updates the partner models achieving greater outcome matrix creation accuracy. The function x maps partner features to a partner ID, y maps situation features to the robot's self model, and z maps partner features to a partner model.

Box 5.2 depicts the algorithm. For clarity, the algorithm is divided into three phases: pre-interaction, interact, and update. During the pre-interaction phase the robot selects models for itself and the partner, calls the Outcome Matrix Creation algorithm constructing the matrix, selects an action and sets its predictions for the interaction. During the interact phase the robot performs the action. Finally, in the update phase, the

robot adjusts its partner model to account for the actual outcome obtained and actions performed.

Interact-and-update Algorithm Example

Input:

Partner features: <axe, oxygen_mask, male>
 Situation features: <fire, smoke, debris>

Pre-Interaction Phase:

Step 1: $m^i = y(\text{axe, oxygen_mask, male})$ where m^i is the robot model of a firefighter assistant with action model $m^i.A^i = (\text{guide-to-victim, alert-fire})$ and utility function $m^i.u^i = ((\text{guide-to-victim, guide-to-victim, -3}), (\text{alert-fire, guide-to-victim, -7}), (\text{guide-to-victim, alert-fire, 2}), (\text{alert-fire, alert-fire, 1}))$

Step 2: Use partner features to retrieve model of partner m^{-i} . If no partner models exist, then assign $m^{-i} = m^i$. Thus, $m^{-i}.A^{-i} = (\text{guide-to-victim, alert-fire})$ and utility function $m^{-i}.u^{-i} = ((\text{guide-to-victim, guide-to-victim, -3}), (\text{alert-fire, guide-to-victim, -7}), (\text{guide-to-victim, alert-fire, 2}), (\text{alert-fire, alert-fire, 1}))$

Step 3: Use models (m^i, m^{-i}) to create matrix. Result is:

Step 4, 5: Set $a^i = \text{guide-to-victim}$; Set $o^i = -3$;
 $* a^{-i} = \text{guide-to-victim}$; $* o^{-i} = -3$

		Robot	
		guide-to-victim	alert-fire
axe-oxygen_mask-male	guide-to-victim	-3	-7
	alert-fire	-3	2

Interact Phase:

Step 6: Perform guide-to-victim action

Update Phase:

Step 7: Perceive values $a^{-i} = \text{perform-CPR}$; $o^{-i} = 4$; $o^i = 9$

Steps 8, 9: Add action perform-CPR to $m^{-i}.A^{-i}$; outcome value (perform-CPR, guide-to-victim, 4) to $m^{-i}.u^{-i}$

Step 12: Add outcome value (guide-to-victim, perform-CPR, 9) to $m^i.u^i$.

Step 13, 14: $p(a^{-i}) > k$ for all a^{-i} assuming $k = 0.1$

Step 15: Goto step 3.

Figure 5.7 This figure presents an example run through the interact-and-update algorithm. The partner and situation features are presented as inputs to the algorithm. In steps 1 and 2 these features are used to retrieve the partner and self models. In step 3, the models are used to create the pictured matrix. Steps 4 and 5 predict actions and outcomes based on the models. Step 6 performs the action and steps 7 through 12 update the models. Steps 13 and 14 delete unused actions, if necessary and Step 15 goes back to Step 3.

The interact-and-update algorithm (Box 5.2) takes as input the partner features and situation features. Partner features are used to recognize and/or characterize the robot's interactive partner. Similarly, situation features are perceptual features used to characterize the environment. The algorithm begins by using the situation features to retrieve a self model. The function y maps situation features to subsets of the robot's action set and utility values. Thus the robot's model of itself depends on the type of environment in which it is interacting. In the example in Figure 5.7 the features of a search and rescue environment self model of a firefighter assistant.

The partner's features are used to retrieve a model of the partner. The function z selects the partner model from a database of partner models with the greatest number of equivalent features. During initialization, the partner model database is seeded with a model of the robot. Thus the database always contains at least one model. In Figure 5.7 this results in the partner model being set to the robot model. The pre-interaction phase also constructs the outcome matrix representing the given situation, selects the action that the robot will perform, and predicts the action that the human will perform, $*a^{-i}$, the outcome value that the robot will receive, $*o^i$, and the outcome value that the human partner will receive, $*o^{-i}$.

During the interaction phase the robot performs the action. In the example presented in Figure 5.7 this is the `guide-to-victim` action.

During the update phase of the algorithm, the robot first perceives the action performed by its partner and the outcome both it and the partner obtain (line 7 from Box 5.2). Next, if the partner action does not match the prediction, then the action is added to the model if it did not exist and the outcome for the action pair is updated (line 8). In

Figure 5.7 this is the case as the robot predicted that the partner would perform the `guide-to-victim` action when in fact they performed the `perform-CPR`. If, on the other hand, the robot predicted the correct action but did not predict the correct outcome then the outcome is updated in the partner model (line 10). Next, if the outcome the robot obtained differed from the robot's prediction then the robot updates its own model to reflect the received outcome (line 12). Finally actions and associated outcome values which have less than k probability of usage based on previous experience are removed. This prevents the model from becoming filled with rarely used actions. The partner model can be successively updated by looping to line 3.

In section 5.2 we saw that outcome matrices are more sensitive to action deletion errors than to action insertion errors. The constant k provides a means of balancing the likelihood of each type of error. A value of $k = 0.5$ deletes actions if they do not have a 50 percent probability of being selected by the partner. This large value of k results in smaller matrices but also result in increased likelihood of action deletion errors. We can reduce action deletion errors by reducing the value of k .

Line 9 updates the outcome value to match the perceived outcome value when an unexpected action is encountered. If the action is unknown, then the robot does not yet have information about the outcome values of all of the action pairs. In this case it must make an assumption as to their value. As currently presented, the algorithm assigns a single outcome value to all action pairs irrespective of the robot's action. This assignment results in what we call an **action independence assumption**. The robot is assuming that, for the unknown action pairs, the partner receives the same outcome regardless of the robot's choice of action. Alternatively, we could have assumed that for unknown action

pairs the human receives the same outcome as the robot. Either of these assumptions is equally valid as the values simply serve as placeholders and allude to the robot's current ignorance of the human's action preference.

We contend that this is neither the only algorithm for matrix creation, nor, perhaps, even the best algorithm for creating and updating outcome matrices. Rather the algorithm is only meant to serve as a starting place for more advanced outcome matrix creation algorithms. Moreover it shows that outcome matrices can be created from perceptual information and demonstrates the connection between a robot's model of its interactive partner and its ability to represent an interaction. Intuitively, the algorithm directly updates the outcome values and actions. Hence the algorithm is susceptible to sensor noise. Machine learning algorithms could be used to reduce this susceptibility. Ng, for example, describes inverse reinforcement learning as the problem of learning a task's reward function. He has also developed techniques for learning from a teacher (Abbeel & Ng, 2004). Numerous game theoretic methods, such as Bayesian games, also exist for handling uncertainty (Osborne & Rubinstein, 1994). The problem of how to best manage uncertainty and noise when constructing outcome matrices for use in human-robot interaction is a challenging question. At this point it is not clear what the nature of the uncertainty will be. For example, are Gaussian noise models appropriate? Will the noise be non-linear? Should game-theoretic or machine learning techniques or both be used to manage noise and uncertainty? This dissertation does not exhaustively probe these questions. Section 5.4, however, does begin to explore the use of clustering methods to aid in partner model learning.

5.3.1 Creating accurate partner models

The purpose of the interact-and-update algorithm is to create outcome matrices. But how accurate are the outcome matrices produced by the algorithm? Accuracy here is defined and measured with respect to a target model. As discussed in section 5.1, the accuracy of

Table 5.8 General experimental information common to all of the experiments performed to investigate the use of the interact-and-update algorithm for the creation of accurate partner models.

General Experiment Summary	
Creating Accurate Partner Models	
Purpose	Determine the ability of the interact-and-update algorithm to create accurate outcome matrices.
Experiment Type	USARSim simulation and laboratory experiments.
Hypothesis	As the number of interactions increases, the accuracy of the robot's partner model will increase.
Procedure	Both procedures are listed within this section: 1) Follow partner model creation procedure from Table 5.12. 2) Follow the experimental procedure from Table 5.13.
Independent variable	Number of interactions with the partner.
Dependent variable	Percent similarity to a target partner model.
Method of Analysis	Target model comparison.

the outcome matrix produced by the General Matrix Creation algorithm depends primarily on the accuracy of the robot's partner and self models. Recall that in section 5.1 we showed that if the robot's partner model and self model are accurate, then the algorithm can be used to create an accurate matrix. We can therefore gauge the ability of interact-and-update algorithm to accurately create outcome matrices by measuring the accuracy of the models created by the algorithm. If the algorithm produces accurate models, then these models can be input into the General Matrix Creation algorithm to produce accurate outcome matrices. Section 4.2 has already presented a method and equations for comparing partner models. To briefly revisit this topic, we examined mechanisms for determining the difference between a robot's model of its partner (m^{-i}) and the partner's actual model ($*m^{-i}$). We noted that our distance measure must include

both the model’s components and Type I and II error. Representing the action model as a set of actions, $a_j^i \in A^i$, and the utility function, u^i , as a set of triplets, $(a_j^i, a_k^{-i}, \mathfrak{R})$, we derived equation (2) from section 4.2 as a measure of partner model distance. In short, we have derived a measure of partner model distance that will now be used to gauge the ability of the interact-and-update algorithm to accurate outcome matrices.

As described in section 5.3, the interact-and-update algorithm operates by successively revising the robot’s partner and self model information. We therefore hypothesized that continued interaction with a partner would result in improved partner model accuracy—both accuracy of the partner’s action model and of the partner’s utility function. Figure 5.7 presents an example interaction between the robot and a firefighter in a simulated search and rescue environment. Continued interaction here means that the robot interacted successively with the same human partner in a single environment.

Table 5.9 Summary of the creating accurate partner models experiment conducted in simulation with a single partner type and in multiple environments.

Experiment Summary	
Creating Accurate Partner Models: Simulation, Single Partner Type, Multi-Environment	
Purpose	Determine the ability of the interact-and-update algorithm to create accurate outcome matrices.
Experiment Type	USARSim simulation involving a single partner type and multiple environment types.
Hypothesis	As the number of interactions increases, the accuracy of the robot’s partner model will increase irrespective of the type of environment.
Procedure	Both procedures are listed within this section: 1) Follow partner model creation procedure from Table 5.12. 2) Follow the experimental procedure from Table 5.13.
Independent variable	Number of interactions with the partner, the type of environment (assistive, museum, household, search and rescue, prison).
Dependent variable	Percent similarity to a target partner model
Method of Analysis	Target model comparison
Conclusion	Hypothesis is supported. Accuracy found to increase with additional interactions irrespective of type of environment.

To test this hypothesis we used USARSim to conduct two simulation experiments. The first experiment examined interaction with a single type of partner (an emergency medical technician or EMT) in each of the five different environments (see section 3.4.1 for environment types). The second simulation experiment explored interaction in a single environment (search and rescue) with four different types of partners: police officer, firefighter, EMT, and citizen. Our motivation in conducting these two simulation experiments was to show that the results are not limited to a particular type of environment or type of partner.

Table 5.10 List of actions available to the robot for each different type of environment.

Robot actions for each different type of Environment	
Environment	Actions
Search and rescue	SearchFor-victim, Observe-victim, Light-victim, GuideTo-victim, SearchFor-victim, Observe-fire, GuideTo-fire, SearchFor-fire
Assistive	SearchFor-patient, Observe-patient, GuideTo-patient
Household	SearchFor-medicine, GuideTo-medicine, SearchFor-homeowner, GuideTo-homeowner, SearchFor-intruder, Observe-intruder, Light-intruder, GuideTo-intruder,
Prison	SearchFor-prisoner, Observe-prisoner, Light-prisoner, GuideTo-prisoner, SearchFor-visitor, Observe-visitor, Light-visitor, GuideTo-visitor
Museum	SearchFor-fire, Observe-fire, GuideTo-fire, SearchFor-intruder, Observe-intruder, Light-intruder, GuideTo-intruder

Table 5.11 A list of actions for each type of partner.

Partner Type	Actions
Police Officer	limit-access, direct-traffic, search-for-victim
Firefighter	remove-toxic-material, fight-fire, rescue-victim, move-debris
EMT	startIV, intubate, performCPR
Citizen	run, cry, scream
Random	Any of the above non-robot actions.

The first simulation varies the type of environment in which interaction occurred. The robot's action models were again environment specific. Thus, each different environment resulted in a different action model and utility function for the robot. In the search and rescue environment, for example, the robot used actions such as SearchFor-victim to help locate trapped victims. In the museum environment, on the other hand, the robot

used actions such as `SearchFor-intruder` in its role as a security guard patrolling the museum. Table 5.10 presents the actions available to the robot in each environment. An arbitrary utility function was also created for each environment.

The robot’s interactive partner in the first simulation experiment was an EMT. It was therefore necessary to create a partner model for an EMT. Table 5.11 presents the action model for the EMT type. An arbitrary utility function was also created for EMT partner type. The following procedure was followed for creating a partner model.

Table 5.12 Procedure for creating partner models.

Partner Model Creation Procedure	
Procedure for creating an individual partner model given the partner type:	
1)	Using a Gaussian distribution, randomly select values for the partner’s features with the exception of the Tool-1 and Tool-2 feature which are type specific.
2)	Use Table 5.11 to set the action model for the partner type
3)	Create arbitrary utility values for the individual.

The simulation experiment involved 20 interactions with the partner in each of the five different environments. An interaction consisted of the performance of an action within the environment by both the robot and the robot’s partner and the observation by the robot of its partner’s action and outcome. The following general procedure was used for the experiment:

Table 5.13 Experimental procedure used for each of the experiments in this section.

Experimental Procedure	
1)	The robot uses OpenCV to detect objects in the environments and creates situation features based on these objects.
2)	The robot uses synthesized speech and speech recognition to query the partner for their features.
3)	The robot now has the information necessary to run the interact-and-update algorithm. As detailed in Box 5.2 and Figure 5.7, the robot uses the situation and partner features to retrieve its self and partner models, constructs a matrix, performs an action from Table 5.10, observes its outcome and its partner’s

action and outcome, and updates its self and partner models. The value for the parameter k is set to 0.10.

- 4) The robot's observation of the partner's action and outcome was accomplished by asking the partner to state the action they performed and outcome that they receive.
- 5) The robot's model of its partner is recorded after each interaction. Equation (1) from section 4.2 is used to calculate the accuracy with respect to the individual components of the partner model. Equation (2) is used to calculate the overall accuracy of the partner model.

Figure 5.8 depicts the results for the first simulation experiment. The graph shows that with continued interaction the accuracy of the action model, utility function, and partner model increase, eventually matching the target model. After the eleventh interaction, the accuracy of all models increases dramatically. This is because the algorithm purges the action model and utility function of seldom used actions and utilities reducing Type I error (mentioned in section 4.2).

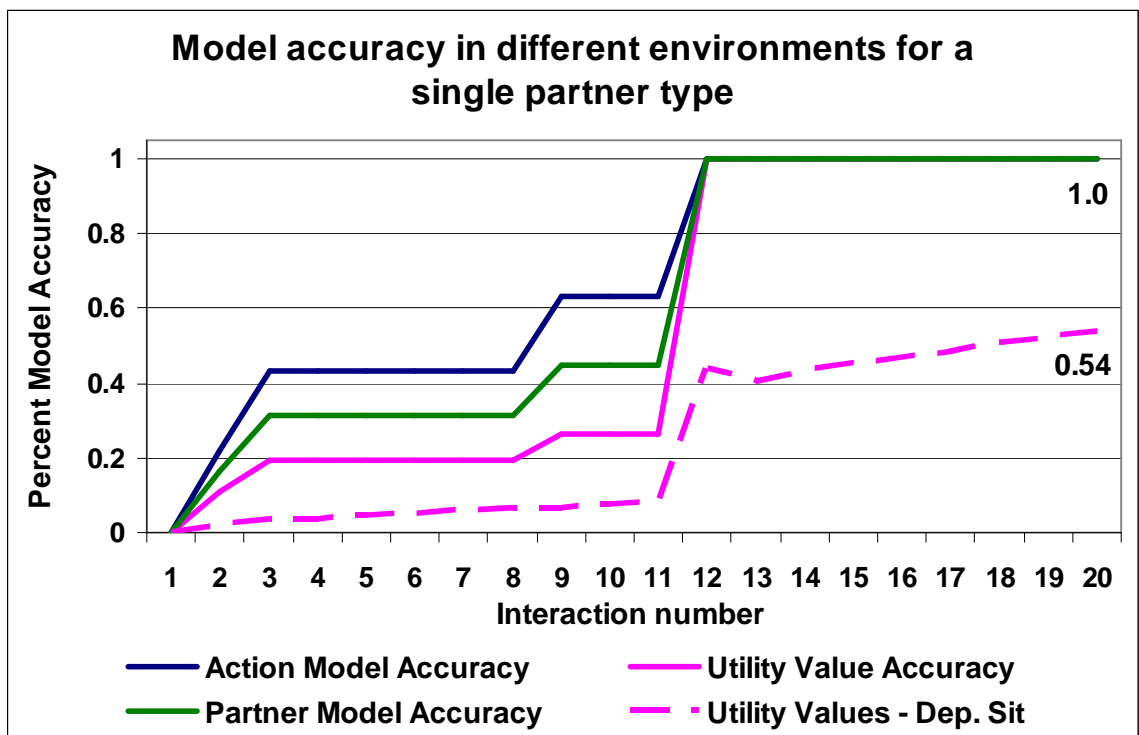


Figure 5.8 The graph depicts the results from the first simulation experiment involving different environments. The results show that model accuracy increases with continued interaction, eventually matching the target model.

Recall that if a situation is independent then the robot and the human receive their respective outcome regardless of the action selected by the other. The action independence assumption holds when the situation is independent—as in the experiment presented above. But what happens when this assumption does not hold? To test this, we reran the above experiment using a dependent situation. In a dependent situation, the outcome received by the robot and the human depends entirely on the action selected by the other. Hence, the use of a dependent situation consistently violates the action independence assumption and should result in poorer performance by the algorithm. We indeed found this to be the case. Figure 5.8 also shows the results when a dependent situation is used. Here we see that accuracy of the utility values only reaches 64% after 20 interactions. This is because the dependent situation violates the action independence assumption discussed in section 5.3. Although less accurate, the partner model in this case still contained all of the information experienced during interaction with the partner. Moreover, because the action independence assumption was violated for every action pair in the matrix, this represents a worst case result.

Creating Accurate Partner Models: Multiple Partner Types, Single Environment

The preceding experiment was limited to a single type of partner. It is important to generalize the results to not just multiple environments, but also to multiple types of partners. In order for the interact-and-update algorithm to be of value, it must work regardless of the information contained within either the self-model (as examined in the previous experiment) or the partner model. Table 5.11 summarizes the experiment.

We again hypothesized that continued interaction with a partner would result in improved partner model accuracy. In this case, however, rather than placing the robot in

different environments, the robot interacted with different types of partners in a single environment. The search and rescue environment was used for this experiment.

Table 5.14 Summary of the creating accurate partner models experiment conducted in simulation with multiple partners and in a single environment.

Experiment Summary	
Creating Accurate Partner Models: Simulation, Multiple Partner Types, Single Environment	
Purpose	Determine the ability of the interact-and-update algorithm to create accurate outcome matrices.
Experiment Type	USARSim simulation involving a multiple partner types and a single type of environment.
Hypothesis	As the number of interactions increases, the accuracy of the robot's partner model will increase irrespective of the type of partner.
Procedure	Both procedures are listed within this section: 1) Follow partner model creation procedure from Table 5.12. 2) Follow the experimental procedure from Table 5.13.
Independent variable	Number of interactions with the partner, the type of partner (EMT, firefighter, citizen, police officer, random).
Dependent variable	Percent similarity to a target partner model
Method of Analysis	Target model comparison
Conclusion	Hypothesis is supported. Accuracy found to increase with additional interactions irrespective of type of partner.

For this experiment, the robot again interacted with an individual twenty times. The partner's features depended, in part, on the partner's type. For example, a police officer could be male or female, tall or short, but, unlike the other partner types, always had a gun and a badge. Hence, two police officers would both have had guns and badges, but one could be a tall male and the other a short female. Table 3.2 provides a list of all partner features. The features named Tool-1 and Tool-2 were again type specific.

Action models were also type specific. Thus, a police officer and a firefighter were both capable of a different set of actions. A police officer, for example, could limit-access to an area, direct traffic, or search for victims. Alternatively, a firefighter could fight a fire, rescue a victim, and move debris.

The utility functions for each individual were unique and arbitrary. Hence, whereas one police officer might prefer to direct traffic over all other actions another could prefer to search for victims.

The target model consisted of predefined sets of actions and outcome values for a specific partner type. For example, a citizen partner was produced by 1) randomly selecting the values for the partner features (except tools which are set to baseball-cap and backpack for this type) 2) setting the action model to that from Table 5.11 for citizen and 3) creating arbitrary utility values for the utility function. This procedure was repeated for each individual partner. A procedure for creating partner models has already been detailed above.

Again simulation experiments involved 20 interactions with a particular individual. The experimental procedure listed above was followed. The robot's model of its partner was again recorded after each interaction.

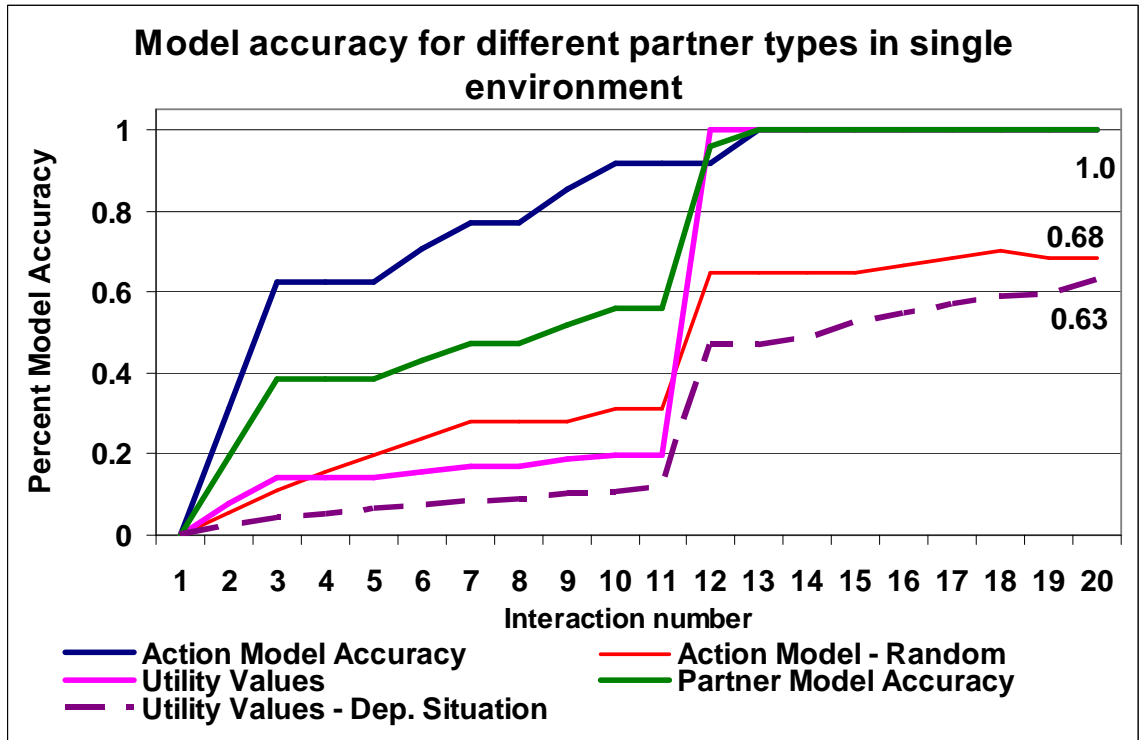


Figure 5.9 The graph depicts the results from the second simulation experiment involving different partner types. The results again show that model accuracy increases with continued interaction, eventually matching the target model.

Figure 5.9 depicts the results for the second simulation. Again the graph shows that the accuracy of all models increases with continued interaction, eventually matching the target model. Violating the action independence assumption again results in decreased utility accuracy (63 percent). A random partner type is also included for comparison. The random partner selected any action available to any partner type at random. The graph only depicts action model accuracy for the random partner type. An accuracy of 68 percent is achieved for the random partner type.

Creating Accurate Partner Models: Laboratory Experiment

A follow-up laboratory experiment was conducting on a Pioneer DX in a mock search and rescue environment. In this experiment the robot was tasked with assisting a firefighter to either rescue victims or to observe the fire. As with the other experiments,

we hypothesized that continued interaction would result in improved model of the partner.

Table 5.15 Summary of the creating accurate partner models experiment conducted in simulation with multiple partners and in a single environment.

Experiment Summary	
Creating Accurate Partner Models: Laboratory Experiment	
Purpose	Determine the ability of the interact-and-update algorithm to create accurate outcome matrices.
Experiment Type	Laboratory experiment conducted in mock search and rescue environment with a Pioneer DX.
Hypothesis	As the number of interactions increases, the amount of outcome obtained by the robot increases.
Procedure	Follow the experimental procedure from Table 5.13.
Independent variable	Number of interactions with the partner.
Dependent variable	Outcome obtained.
Conclusion	Hypothesis is supported. The amount of outcome obtained by the robot increases.

In this experiment, the robot's action model consisted of two actions: 1) moving to and observing a victim and 2) moving to and observing a hazard. The robot received more outcome if the victims survived. The victims survived only if the robot and the firefighter work together observing and containing the hazard or rescuing the victims (Figure 5.10 image 4 shows the victims and hazards). Hence, for the robot, task performance depended on the accuracy of its model of the partner.

The robot's partner also choose among two potential actions: 1) containing the hazard or 2) rescuing the victims. The firefighter arbitrarily preferred to contain hazards. Hence, the human's utility function showed a preference for containing the hazard.

Both the robot and the human select the actions concurrently. The same experimental procedure used in the two preceding experiments (presented in Table 5.13) was again used for this experiment.

The experiment consisted of five interactions. Initially the robot has no knowledge of the action model or utility functions of its partner. The robot therefore sets its partner model to the robot's self model. In other words the robot assumes that its unknown partner has the same actions and preferences that it does. During the first interaction the robot moves to observe the victims falsely believing that the firefighter will also move to rescue the victims. After the interaction, the robot receives feedback indicating that the firefighter moved to contain the hazards. It updates its partner model accordingly and during the next interaction it correctly moves to observe the hazard. Figure 5.10 images 1-3 show the robot moving to observe the victim in the first interaction and the hazard in the second interaction. Figure 5.10 image 5 depicts the video sent by the robot to the human.

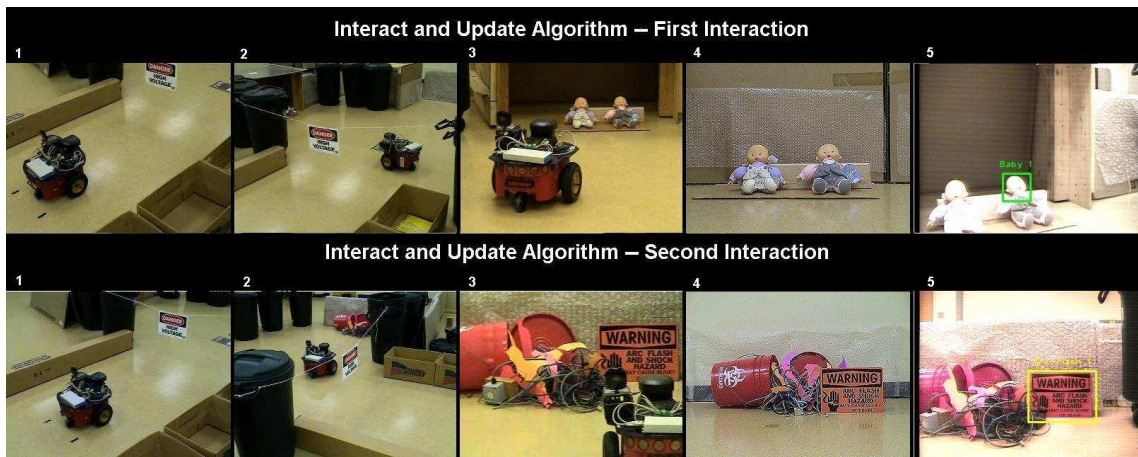


Figure 5.10 Photos from the robot experiment. The robot initially moves to observe the victim. After learning the model of its partner the robot moves to observe the hazard. Photos 1-4 depict the robot as it moves through the maze and selects actions. Photo 5 depicts video that the robot sends to its human partner.

Because the robot does not initially coordinate its behavior with the firefighter, it obtains an outcome of zero. Afterward and for the remaining interactions, the robot obtains an outcome of two (because two victims are saved). A net increase of two victims per interaction eight total victims' results from the robot's modeling of the partner.

This experiment is merely meant to demonstrate the potential feasibility of this approach on a robotic platform. As a pilot study, the results show that algorithm operates on a robotic platform in a laboratory environment, and as such, should allow for more extensive testing in more realistic environments.

5.3.2 Interact and Update algorithm conclusions

This section has introduced an algorithm that, when combined with the General Matrix Creation algorithm, produces outcome matrices which represent a robot's interaction with its human partner. The algorithm, however, assumes perceptual competencies which are difficult to achieve given the current state of the art. It assumes that the robot can perceive 1) the partner's action, 2) the partner's outcome value, and 3) the outcome obtained by the robot itself. These assumptions may limit the current applicability of the algorithm. Nonetheless, as demonstrated by experiments, the perceptual limitations of this algorithm can be circumvented. Moreover, activity recognition and affect detection are current areas of active research (Philipose et al., 2004; Picard, 2000). Finally, it is important that the HRI community recognize the importance of activity recognition and state detection. This research provides a theoretical motivation for these research topics. It may well be that the challenge of recognizing how a robot's behavior has impacted the humans interacting with the robot is a critical question facing the HRI community.

We have also assumed that the robot knows what actions are available to it. We believe that this is a reasonable assumption. We have not assumed that the robot has accurate knowledge of the outcomes values resulting from the selection of an action pair. We have simply assigned arbitrary initial values for the outcomes and then the robot learns the true values through interactive experience with the partner.

Although our results show that interactive experience creates increasingly accurate partner models, the actions and utilities of the robot's partner were static and contained no noise. Because the models were static they could be modeled. Alternatively, as demonstrated in the random partner type, the partner could have continually selected random actions or received random utilities. Clearly in this case less can be learned about the partner. In a sense, the robot cannot know what to expect next from its partner. In normal interpersonal interaction there are times when humans randomize their interactive actions, such as in some competitive games. This algorithm will have limited success in these situations. Noise in the form of inaccurate perception of the human's outcome values and actions is another potential challenge. Fortunately, game theory provides numerous tools for managing outcome uncertainty (Osborne & Rubinstein, 1994). Moreover, the results presented in section 5.2 have demonstrated that outcome matrices degrade gracefully with increased error (Wagner, 2008). Future work may employ machine learning and/or game-theoretic techniques to reduce overfitting.

Near-term practical applications of this work would likely focus on environments where the outcomes of the robot's partner are readily available. In assistive therapy environments, for example, the robot could ask the patient if an exercise was causing pain. An entertainment robot, on the other hand, might gauge user outcome in terms of amount of time spent interacting with the robot. Applications in areas such as autism are more difficult because the nature of the disease may limit the human's outcome expression capabilities.

Neuroscientists have shown that humans actively model their interactive partners (Rilling, Sanfey, Aronson, Nystrom, & Cohen, 2004). Certainly the interpersonal mental

models maintained by humans are more complex and rich than the models used here. Our purpose is not to claim that the partner models discussed here are the same as those formulated by humans, but rather to explore what minimal modeling of its interactive partner a robot must perform in order to interact successfully with the partner and to present a method for achieving this modeling. The section that follows introduces a method by which the robot can learn and generalize from collections of partner models, reducing the number of interactions needed to model its partner.

5.4 The Stereotype Matching Algorithm

Psychologists note that humans regularly use categories to simplify and speed the process of person perception (Macrae & Bodenhausen, 2000). Macrae and Bodenhausen suggest that categorical thinking influences a human's evaluations, impressions, and recollections of the target. The influence of categorical thinking on interpersonal expectations is commonly referred to as a stereotype. For better or for worse, stereotypes have a profound impact on interpersonal interaction (Bargh, Chen, & Burrows, 1996; Biernat & Kobrynowicz, 1997). Information processing models of human cognition suggest that the formation and use of stereotypes may be critical for quick assessment of new interactive partners (Bodenhausen, Macrae, & Garst, 1998). From the perspective of a roboticist the question then becomes, can the use of stereotypes similarly speedup the process of partner modeling for a robot?

This question is potentially critical for robots operating in complex, dynamic social environments, such as search and rescue. In environments such as these the robot may not have time to learn a model of their interactive partner through successive interactions.

Rather, the robot will likely need to bootstrap its modeling of the partner with information from prior, similar partners. Stereotypes serve this purpose.

Before detailing our algorithm for stereotype learning and use, we must first define our terms. Sears, Peplau and Taylor define a stereotype as an interpersonal schema relating perceptual features to distinctive clusters of traits (Sears, Peplau, & Taylor, 1991). With respect to our framework, then, a stereotype is a type of generalized partner model used to represent a collection or category of individual partner models. Thus, the creation of stereotypes requires the creation of these generalized partner models. Moreover, to be useful, stereotypes must be matched to the partner's perceptual features. Stereotype building will therefore be a two phase process. First, we cluster partner models with the centroids of the clusters becoming the partner model stereotype. Next, we learn a mapping from partner features to the stereotypes. Our implementation utilizes agglomerative clustering and C4.5 decision trees (Quinlan, 1994). We conjecture that the algorithm will work for any type of clustering algorithm and machine learning algorithm, but do not offer evidence to support this statement. Box 5.3 details stereotype creation and Box 5.4 describes how to use a stereotype.

Stereotype Matching Algorithm: Building Stereotypes

Input: Partner Model m^{-i}

Output: Classifier ψ mapping m^{-i} .features to a stereotype.

```

Cluster phase //the cluster phase clusters models to build stereotypes
1.   Add  $m^{-i}$  to partner model space //the partner model space is a set of partner models
2.   for all models in model space
3.       make a cluster
4.   while centroid_distance( $c_j, c_k$ ) <  $k$ 
5.       merge_clusters( $c_j, c_k$ )
Function learning phase //this phase maps stereotypes to partner features
6.   for all models in model space
7.       set data[i]  $\leftarrow$  make_pair( $m^{-i}$ .features, cluster centroid)
8.    $\psi \leftarrow$  build_classifier(data)
9.   return  $\psi$  //return a classifier mapping features to stereotypes
    
```

Box 5.3 Our algorithm for stereotype creation. The algorithm takes a new partner model as input. It then creates clusters of all of the stored models. The cluster centroids will serve as the robot's partner stereotypes. In the function learning phase, the robot learns a mapping from partner's features to the stereotypes. This mapping can now be used to retrieve a stereotype given the partner's perceptual features.

Stereotype Matching Algorithm: Using Stereotypes

Input: Partner features $f_1^{-i}, \dots, f_n^{-i}$

Output: Partner model m^{-i} .

```

1.   If classifier == null //if we have not built the classifier then return
2.       return null
3.   convert  $f_1^{-i}, \dots, f_n^{-i}$  to instance of classifier data
4.   result  $\leftarrow$   $\psi$ .classify(instance) //use the features as input to the classifier
5.    $m^{-i} \leftarrow$  stereotypeList(result) //once the stereotype is known, return the
6.   return  $m^{-i}$  // partner model for that stereotype
    
```

Box 5.4 The stereotype matching algorithm uses the partner's features to retrieve a stereotyped partner model.

As depicted above, the building stereotype algorithm takes as input a new partner model. This input is optional. The stereotype building algorithm can also be run on the robot's existing history of partner models (termed the model space). The interact-and-update algorithm is used to create the models that occupy the model space. The first step

of the algorithm adds the new model to the model space. Next each model in the space is assigned to a unique cluster. The third and fourth steps perform agglomerative clustering, iterating through each cluster and, if the clusters meet a predetermined distance threshold, merging them. Equations (1) and (2) for partner model distance, which were first presented in section 4.2 and briefly reviewed in section 5.3, are used to determine if the clusters are meet the predetermined distance threshold for merging. The cluster centroids that remain after step four are the stereotypes, denoted s_1, \dots, s_n . A list of stereotype models is kept by the robot.

In the next phase we use clustering to create a function, ψ , mapping the partner's perceptual features to the stereotype. Line 7 from Box 5.3 creates data for the machine learning algorithm by pairing each model's perceptual features to a stereotype. In the final steps, this data is used to train a classifier mapping partner features to the stereotyped model. Figure 5.11 presents an example.

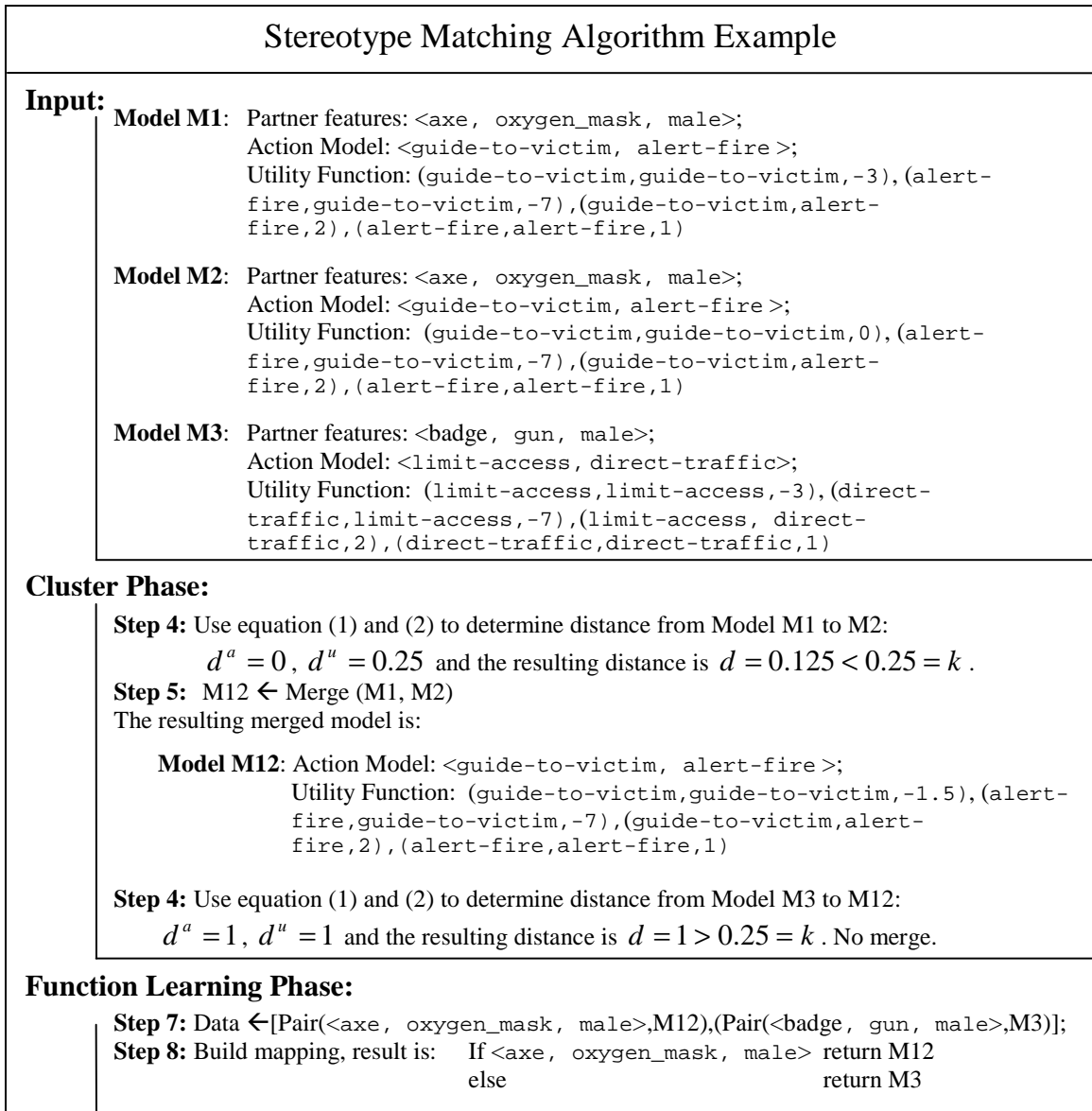


Figure 5.11 An example run of the stereotype building algorithm. Three partner models serve as input to algorithm. In the cluster phase, the algorithm first merges models M1 and M2 creating model M12. The distance between model M3 and M12 is greater than 0.25 so the model is not merged into the stereotype. In the function learning phase the stereotype clusters are paired with the partner features. A classifier is constructed from the resulting data. The example skips some of the most easily understood steps, such as the for loops.

The stereotype building algorithm makes two important assumptions. First, it assumes the existence of a distance function, $d(m_i^{-i}, m_j^{-i})$, capable of measuring the difference between two partner models. We have already described our method for measuring partner model distance (see section 4.2). If, however, additional information (such as the

partner's beliefs, motivations, goals, etc.) is added to the partner model, then creating a distance function may become difficult because this information may not naturally have a measure for determining distance. Second, the stereotype building algorithm assumes that partner models can be merged to create new partner models. In order to merge a partner model one must merge the components of the partner model. For this work that meant merging the action models and utility functions. Action models were merged by adding an individual action only if the action was included in half of the data that composed the merged model. For example, if the merged model was created from ten individual partner models and an action existed in four of the models then it was not included in the merged model. If, however, the action existed five of the models then it was included in the merged model. Similarly, merged utility values were derived from the average utility value of the composition utility functions. Again this is not the last word in either gauging the distance between partner models or in merging models. This work, however, does represent, to the best of our knowledge, the first time that a robot has used stereotypes to guide its interactive behavior.

To use a stereotype the robot simply converts the partner's features into an instance of data for the classifier and then uses the classifier to select the correct model (Box 5.4).

One important question is how the algorithm reacts to partners that conflict with its stereotypes. For example, a new partner with features resembling a police officer would cause the algorithm to retrieve a stereotype created from the merged models of all other police officers encountered by the robot. This retrieved model allows the robot to predict a particular action model and utility function. If, while interacting with the partner, the robot finds that its action model and/or utility function are inaccurate (i.e. if it predicts

incorrect actions or utilities) then the interact-and-update algorithm alters the current partner model to reflect the differences. Finally the new model is added to the model space and when the build stereotype algorithm is run again this model is included in the stereotype generation process. The end result is that the robot’s stereotype of police officer becomes more general and less specific with regard to the stereotype’s actions and utilities.

5.4.1 Examining the use of stereotypes

As mentioned in the previous section, psychologists claim that human use of stereotypes allows for quicker assessment of new interactive partners (Macrae & Bodenhausen, 2000). We hypothesized that the use of stereotypes by a robot would require fewer interactions to obtain equal partner model accuracy when compared to the interact-and-

Table 5.16 Summary of the use of stereotypes experiment conducted in simulation with multiple partners and in a single environment.

Experiment Summary Examining the use of stereotypes: Simulation Experiment	
Purpose	Investigate the possibility of learning and using clustered partner models, stereotypes, to speedup the process of partner modeling.
Experiment Type	USARSim simulation.
Hypothesis	The use of stereotypes requires fewer interactions to obtain equal partner model accuracy when compared to the interact-and-update algorithm alone.
Procedure	Both procedures are listed within this section: 1) Follow partner model creation procedure from Table 5.12. 2) Follow the experimental procedure from Table 5.17.
Independent variable	Number of interactions with a partner; Number of partners.
Dependent variable	Percent similarity to a target partner model
Method of Analysis	Ablation experiment consisting of comparison to interact-and-update algorithm without use of stereotypes. Target model comparison.
Conclusion	Hypothesis is supported. Fewer interactions are required to obtain equal partner model accuracy when using the stereotype matching algorithm.

update algorithm alone. The experiment we conducted is an ablation experiment comparing the performance of the interact-and-update algorithm with and without the use

of stereotypes. The stereotype matching algorithm allows the robot to learn, generalize, and store information about the partners with which it has interacted. Hence, we believe the algorithm will bootstrap the process of creating an accurate partner model. We reasoned that the stereotypes, even if not perfectly accurate, would still provide useful information that could later be refined by the interact-and-update algorithm. If our hypothesis is correct, then we expect that when the robot encounters a new partner the use of stereotypes will aid in its modeling of this partner and hence result in greater partner model accuracy in early interactions. Table 5.16 provides a summary of the experiment.

To test this hypothesis, we again conducted both simulation experiments and real robot experiments. The simulation experiment had two conditions: using the stereotype matching algorithm (experimental condition) and not using the stereotype matching algorithm (control condition). In both conditions the robot interacted twenty times with twenty different partners. Hence a total of 400 interactions occurred. The partner features, action models and utility functions were identical in both conditions. Moreover, the robot encountered the partners in the same predetermined order in both conditions (see Table 5.18 for order).

In the control condition the robot used the interact-and-update algorithm to gradually build models of each of the partners. Figure 5.7 presents an example run of the interact-and-update algorithm.

In the experimental condition, the stereotype matching algorithm (Box 5.3 and Box 5.4) were used to create stereotypes and to match each partner to an existing stereotype, if one existed. The stereotype matching algorithm can be run in conjunction with the

interact-and-update algorithm or separately. As the interact-and-update algorithm creates each new partner model, the model is used as input to the stereotype matching algorithm. The robot was not given any a priori information related to the stereotypes, such as how many stereotypes to construct.

Four different partner types were again created: a police officer type, a firefighter type, an EMT type, and a citizen type. Each of the robot's different interactive partners was randomly generated from one of the four different types. For example, as depicted in Table 5.18, the robot first interacts with a firefighter for twenty interactions, then an EMT for twenty interactions, a police officer, and so on. The target model consisted of predefined sets of actions and outcome values for a specific partner type. As in the previous experiment, partner feature vectors consisted of values for gender, height, age, weight, hair color, eye color and two objects the individual possessed. Table 5.11 lists the actions available to each type of partner. For this experiment, however, we generated an equal number of each type of partner, randomized the order in which the robot interacted with the different partners, and introduced random differences to the models. In order to ensure that the firefighter partner model, for example, did not always contain the same actions and utilities, randomized differences were introduced to the models. These changes assured that the robot did not interact with individuals that always perfectly reflected the stereotype. The procedure from Table 5.12 was used to create the partner models.

The simulation experiment was conducted in the search and rescue environment (Figure 3.4). Table 5.11 lists the robot's action model. The robot was given an arbitrary utility function.

The following experimental procedure was followed:

Table 5.17 Experimental procedure used in the examining the use of stereotypes simulation experiment. The experiment compares an experimental condition to a control condition. Steps 2 and 4, therefore, only occur in the experimental condition.

Experimental Procedure

- 1) Procedure from Table 5.12 used to construct a target model.
- 2) **Experimental condition:** For each new partner, the using stereotypes algorithm is used to bootstrap the partner modeling process (Box 5.4).
- 3) The procedure from Table 5.13 for the interact-and-update algorithm is followed resulting in partner model m^{-i} . The robot interacts with each partner twenty times.
- 4) **Experimental condition:** After twenty interactions, the partner model m^{-i} is used as input to the stereotype building algorithm (Box 5.3).
- 5) The robot's model of its partner is recorded after every interaction. Accuracy was again determined by comparing the percentage of actions and utilities that were in both the robot's partner model and the target model for the partner (see section 4.2 for details).

The independent variable in this experiment was the use or lack of use of the stereotype matching algorithm. The dependent variable consisted of partner model accuracy.

Figure 5.12 shows the results for the experiment. The x -axis depicts the interaction number and partner number (P0-P19) throughout the experiment. The solid red (dark gray) lines depict a running average of the control condition. As expected, the accuracy of the robot's partner model is consistently poor when interacting with a new partner and results in the regular wave like pattern (red/dark gray). Because the robot does not learn across partners, it must rebuild its partner model with each new partner. Hence, with each new partner the robot's model is inaccurate until it gradually learns about the partner through interaction.

Accuracy of partner model as a function of interactions with different partners

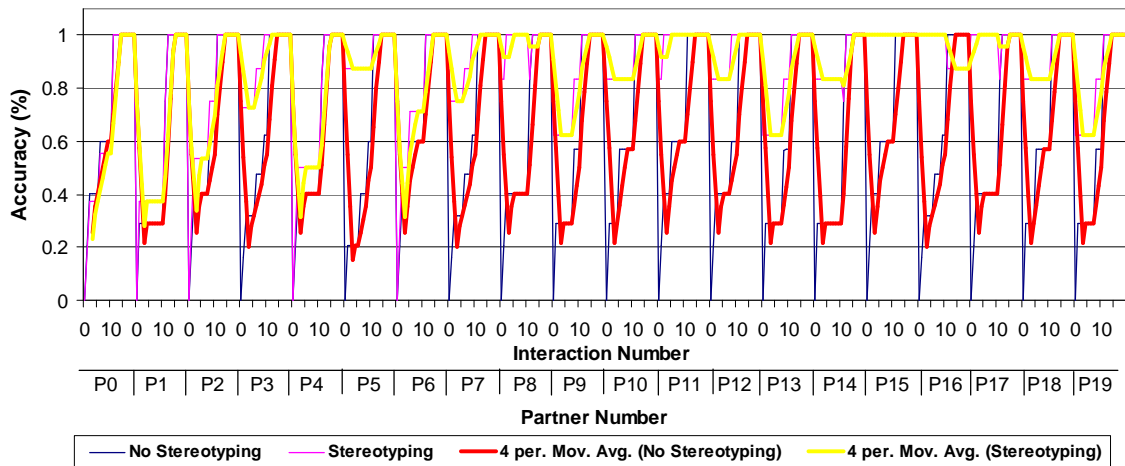


Figure 5.12 Results from the use of stereotypes simulation experiment are depicted above. The bold red (darker gray) line indicates is a moving average for the no stereotyping condition. The bold yellow (light gray) line is a moving average for the stereotyping condition. Stereotyping requires fewer interacts to obtain an accurate partner model once the stereotypes have been constructed. Prior to stereotype construction, however, both methods perform the same. Note that the accuracy of the yellow (light gray) line does not decrease as much as the red line for later partners (P7-P19).

In the experimental condition (yellow/light gray line), however, we see that learning and using stereotypes eventually aids the robot’s performance. Initially the robot has no stereotype information. Hence its performance is equal to the no stereotype condition during P0, P1, P2, and P4. It must learn this information from its interactions with the different partners. The first several partners (specifically P0, P1, P2, P4, and P6) result in continued refinement of the robot’s stereotype models. This occurs as the robot constructs clusters that reflect the different partner types and a decision tree mapping the partner’s perceptual features to these clusters (Table 5.18). After the seventh partner the robot has interacted with enough different partners to have stereotype models for each partner type. In this case, the stereotype model has 80 percent of the same values (actions and utilities) as the partner model. For the remaining partners (P8-P19) the stereotype models only need slight changes (missing action or inaccurate utility value) in order to reflect the partner’s actual model. This fact is shown by the relatively high level of

performance depicted by the yellow (light gray) line in the later interactions. Using 80 percent accuracy as a threshold, the control condition requires an average of 10.2 interactions to reach this threshold. The experimental condition using stereotypes, on the other hand, required only 4.45 interactions on average. This result is significant ($p < 0.01$).

Table 5.18 This table depicts the change in number of clusters and decision tree structure as the robot progressively interacted with different partners during the experiment. We see that by the seventh partner the robot has created clusters for each type. Moreover, after interacting with this seventh partner the robot’s decision tree accurately assigns a stereotype model based on the partner’s perceptual features (Figure 5.12).

Cluster and Classifier Progression with each Partner			
Partner Number	Partner type	Number of Clusters	Decision Tree After Interaction
P0	Firefighter	0	fire
P1	Police Officer	1	fire
P2	EMT	2	fire
P3	Firefighter	3	if (hair=blonde)→police; else fire
P4	Police Officer	3	if (tool1=axe) →fire; else if (tool1=gun)→police else if (tool1= stethoscope)→doctor; else fire
P5	Firefighter	3	if (tool1=axe) →fire; else if (tool1=gun)→police else if (tool1= stethoscope)→doctor; else fire
P6	Citizen	4	if (tool1=axe) →fire; else if (tool1=gun)→police else if (tool1= stethoscope)→doctor; else citizen
P7	Firefighter	4	“
P8	Police Officer	4	“
P9	EMT	4	“
P10	Citizen	4	“
P11	Citizen	4	“
P12	EMT	4	“
P13	EMT	4	“
P14	Police Officer	4	“
P15	Citizen	4	“
P16	Firefighter	4	“
P17	Police Officer	4	“
P18	Citizen	4	“
P19	EMT	4	“

Table 5.18 details the ordering of the partner types that the robot interacted with. As the robot interacts with each different type it adds clusters. Moreover, as shown in Figure 5.13, the robot’s mapping from perceptual features to stereotype model becomes more

accurate with additional training data (partners). Figure 5.13 graphs the accuracy of the classifier with respect to additional partners for the preceding experiment. The graph shows that additional training data in the form of interactive partners increases classifier accuracy. The classification accuracy goes to 100 percent because the partner's features were spoken and no artificial noise was added. In fielded systems the accuracy of the classifier will certainly decrease. The fact that the classifier accuracy goes to 100 percent indicates that the classifier correctly selects a stereotype model when given perceptual features. It does not mean that the model accurately reflects the partner.

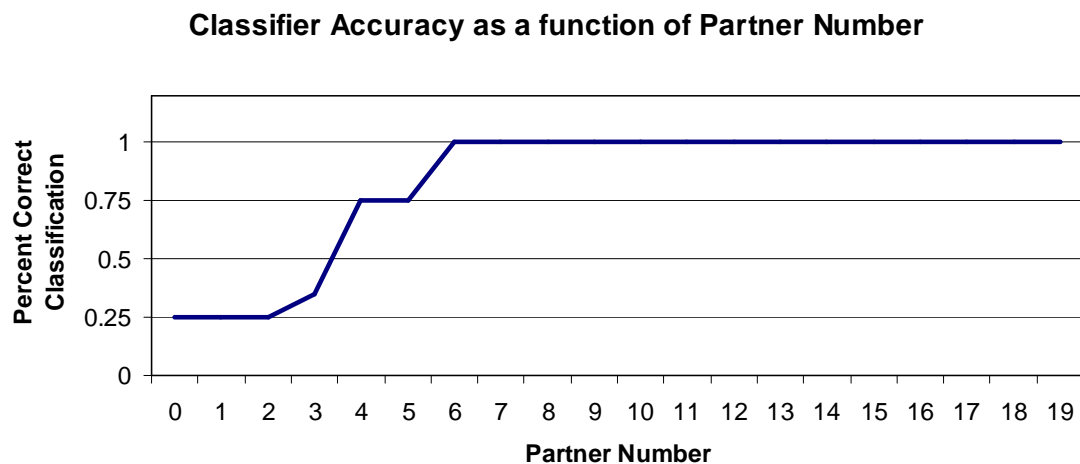


Figure 5.13 The graph depicts the accuracy of the classifier mapping a partner's perceptual features to a stereotype model. As the robot interacts with additional partners the classifier has additional training data and its accuracy increases. The fact that the classifier accuracy goes to one indicates that the classifier correctly selects a stereotype model when given perceptual features. This does not mean that the model accurately reflects the partner.

As a side note, the classifier that emerges from interaction with several different partners (see Table 5.18) only uses a single perceptual feature (from eight possible features) to select a stereotype for the partner. The classifier could potentially be used as a feature selection function, eliminating partner features which do not have any bearing on the partner model. For example, when interacting with five different firefighters the

classifier encounters partner features for both male and female firefighters. Hence, the classifier learns that the gender feature does not reliably map to the firefighter stereotype.

Creating Accurate Partner Models: Laboratory Experiment

A follow-up laboratory experiment was conducting on a Pioneer DX in a mock search and rescue environment. In this experiment the robot was again tasked with assisting a firefighter to either rescue victims or to observe the fire. Here we hypothesized that use of the stereotype matching algorithm would result in additional outcome (task performance) on the part of the robot.

Table 5.19 Experimental summary for the laboratory experiment relating to the use of stereotypes.

Experiment Summary	
Examining the use of stereotypes: Laboratory Experiment	
Purpose	Investigate the possibility of using clustered partner models, or stereotypes, to select the improve task performance.
Experiment Type	Laboratory experiment conducted in mock search and rescue environment with a Pioneer DX.
Hypothesis	The use of stereotype matching algorithm results greater outcome obtainment (task performance) than not using the stereotype matching algorithm.
Procedure	Follow the experimental procedure from Table 5.13.
Independent variable	Control or experimental condition.
Dependent variable	Number of victims saved.
Method of Analysis	Statistical significance (t-test) of number of victims saved.
Conclusion	Results were not statistically significant.

In this experiment, in contrast to the simulation experiments, the robot did not learn the stereotypes because learning of the stereotypes required approximately 20 interactions whereas the robot’s battery life was well below 20 interactions. Rather, the robot was provided with two stereotypes (a firefighter and an EMT) and used perceptual information about the partner to select the correct model and perform the correct action. These stereotypes accurately reflected the partner model for each type, including the correct actions available and utility functions.

If the robot's partner was of type firefighter then the partner's action model consisted of either containing the hazard or rescuing the victims. If, on the other hand, they were of type EMT their action model consisted of either starting an IV or performing CPR. The firefighter arbitrarily preferred to contain hazards and the EMT arbitrarily preferred to perform CPR.

The robot's action model consisted of either moving to and observing a victim or moving to and observing a hazard. The robot received more outcome if the victims survived. The victims survived only if the robot and the firefighter work together observing and containing the hazard or rescuing the victims (Figure 5.14 image 4 shows the victims and hazards). Hence, for the robot, task performance depended on the accuracy of its model of the partner.

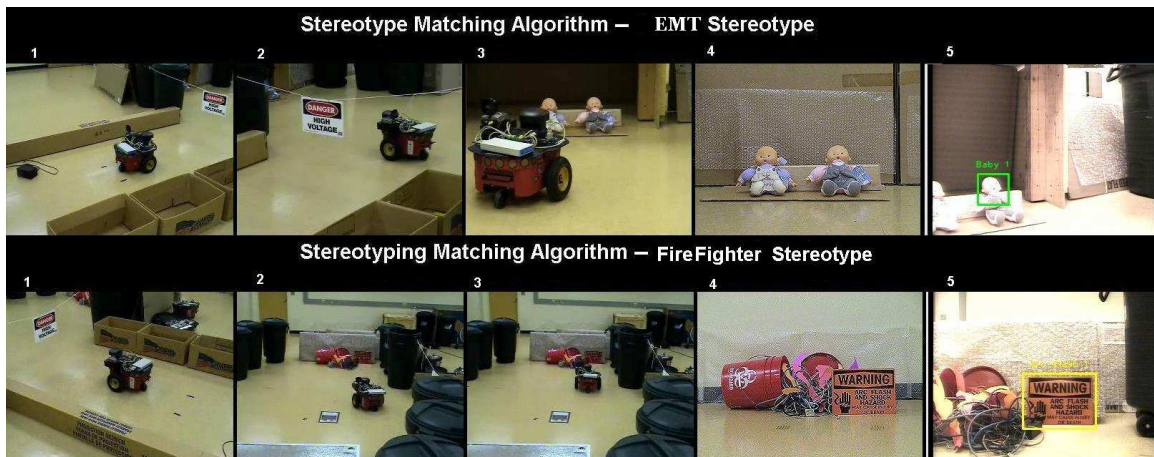


Figure 5.14 The photos above depict the robot using stereotypes to select the correct partner model and then performed an action in a notional search and rescue environment. The first three photos depict the robot performing the action. The next two depict the targets and the robot's view of the targets. When interacting with a person with the perceptual features of an EMT the robot retrieves the EMT stereotype model from memory. It uses this model to determine which of its actions the EMT would prefer and then does that action. The same is true for the firefighter.

Both the robot and the human select the actions concurrently. The same experimental procedure used in the interact-and-update experiments (presented in Table 5.13) was

again used for this experiment with the modification that the robot used the algorithm in Box 5.4 to retrieve a stereotype model for the partner.

The experiment consisted of six interactions both with and without the use of stereotypes. In the experimental condition (use of stereotypes) the robot uses the partner features to retrieve the correct stereotype model. If the model is that of the firefighter, then the robot uses the information within the partner model to construct an outcome matrix indicating that the partner's preferred action is to contain the hazard. The robot therefore selects the `observe-hazard` action to obtain maximal outcome (Figure 5.14 firefighter sequence). If, on the other hand, the stereotype of the EMT is retrieved, then the robot constructs an outcome matrix indicating the EMT's preference to perform CPR and recognizes that it can best help by selecting the `observe-victim` action (Figure 5.14 EMT sequence). A Gaussian distribution was used to randomly determine whether the robot would interact with an EMT or a firefighter. In the control condition, the robot uses a Gaussian distribution to randomly select its action.

All experimental trials resulted in retrieval of the correct stereotype. A total of twelve victims were rescued over all six experimental trials. A total of eight victims were saved over all six control conditions. This result was not significant ($p \approx 0.145$).

Overall, this experiment demonstrates the use of the stereotype matching algorithm (Box 5.4) on real robots in a laboratory environment. The lack of significance is a reflection of the small number of trials conducted.

5.4.2 Stereotype matching conclusions

This section has demonstrated that the use of stereotyped partner models can bootstrap the process of learning a model of the robot's interactive partner. The algorithm we have

presented clusters individual partner model information to create generic or stereotyped partner models that the robot can then use during its initial interactions with a new partner. We presented experiments showing that use of these stereotyped models aids during early interaction with a new partner. Overall, the use of stereotypes may be a natural and important method for the robot to use when it encounters unfamiliar individuals and social situations. Clearly further experimentation is necessary in naturalistic environments with differing partners in order to verify the value of these methods.

The stereotype algorithm has assumed that actual, learnable patterns of partner characteristics exist in the social environment. Psychological literature indicates that this is the case and that humans regularly use this information to categorize and make predictions about their own interactions (Bargh, Chen, & Burrows, 1996; Biernat & Kobrynowicz, 1997). Again, questions of sensor noise in feature and action detection can be raised. Neither of our experiments purposefully injected artificial noise into the system to examine fault tolerance. Moreover, our use of speech recognition for partner feature detection and action and outcome perception resulted in no perceptual noise. Hence the scalability of these methods when faced with significant error and noise is still an open question.

There may be ethical concerns as to whether or not a robot should be empowered with the ability to create and use stereotypes. Our position on this topic is that stereotypes, whether warranted or not, is just another form of human social learning and that in order to best understand this phenomena we must use all of the tools available to explore it. Hence, imbuing robots with the ability to stereotype their human partner, may

allow us to better understand the psychology of stereotypes by offering plausible computational methods by which the phenomena could possibly be realized in the brain.

5.5 Creating Outcome Matrices: Conclusions

This chapter has begun to tackle the difficult question of how to create outcome matrices from a robot's perceptual information. We began by presenting the General Matrix Creation algorithm. This algorithm simply populated the outcome matrix with information, but, importantly, demonstrated that the question of how to create an outcome matrix can be restated as a question of how to create accurate partner models. The interact-and-update algorithm, therefore, was created as a method for both generating outcome matrices and refining the robot's model of its partner simultaneously. The interact-and-update algorithm uses a robot's interactive experience to continuously refine its model of its interactive partner. It, however, did not include methods for learning and generalizing across partners. Hence, we presented the stereotype matching algorithm for this purpose. The stereotype matching algorithm clusters the partner models the robot has learned and uses the cluster centroids as a generalized partner model representing a class of interactive individuals.

Admittedly, we have only begun to address the use these algorithms in real world environments with normal people as the users. Detailed examinations of the type and nature of the noise and uncertainty faced by robots in these situations will be necessary before any definitive judgment can be made as to their efficacy. For the purpose of this dissertation, we have merely attempted to show that it is possible to create our representation of interaction. With respect to the research questions posed in the first chapter, we have shown that a robot can represent interaction and that this representation

can be created from perceptual information obtained by the robot. In the sections that remain, we will show that this representation can have an important impact on a robot's ability to select social actions, represent its relationships, and reason about trust.

CHAPTER 6

SITUATION ANALYSIS

Sociologists and social psychologists have long recognized the importance of the situation as a determining factor of interpersonal interaction (Kelley et al., 2003; Kelley & Thibaut, 1978; Rusbult & Van Lange, 2003). Solomon Asch, a renowned psychologist, stated that, “most social acts have to be understood in their setting and lose meaning if isolated.” (Kelley & Thibaut, 1978). If a goal of artificial intelligence is to understand, imitate, and interact with humans then researchers must develop theoretical frameworks that will allow an artificial system to, (1) understand the situation-specific reasons for a human’s social behavior, and (2) consider the situation’s influence on the robot’s social behavior. Understanding human interactive behavior is critical as it implies that the robot will then be capable of predicting and planning for future interactions and their consequences. Recognition of the situational impacts on a robot’s own interactive behavior is similarly necessary if robots will be expected to operate in the presence of humans in social settings such as the home or the workplace.

This chapter contributes an algorithm for extracting situation-specific information and uses this information to guide interactive behavior. For our purposes, a social situation describes the environmental factors, outside of the individuals themselves, which influence interactive behavior. The objectives of this chapter are to 1) present a novel algorithm for situation analysis developed by the author from interdependence theory that provides a robot with information about its social environment; and 2) demonstrate that the algorithm provides information that can be profitably used to guide a

robot's interactive behavior in certain circumstances. Simulation experiments accomplish these objectives. These simulations first demonstrate that the algorithm is applicable to robotics problems involving collaborations among humans and robots and then examine the algorithm's effectiveness across a wide expanse of social situations.

Consider, as a running example, an industrial accident involving a toxic spill and injured victims. A teleoperated robot is assigned to rescue victims and an autonomous robot operates simultaneously to cleanup the spill. During the cleanup, both the human and the robot will select behaviors directed towards the effort. Perhaps, due to the properties of the spilled material, the victims need to be cleaned before being rescued. In this case, the success of the cleanup depends entirely on both robots working together. Alternative chemical spills will allow the robot and the human to operate in an independent manner, with victims being rescued separately from the cleanup. In either case, the situation should influence the autonomous robot's decision to coordinate its cleanup behavior with the human or to operate independently. Moreover, the effectiveness of the cleanup will depend on the robot's ability to characterize the situation and to use this characterization to select the appropriate behaviors.

The remainder of this chapter begins by first summarizing related research. Next, our algorithm is described, followed by a set of experiments used to examine the algorithm. This chapter concludes with a discussion of these results and directions for future research.

6.1 Situation-based Human-Robot Social Interaction

Interdependence theory underlies our framework for situation-based human-robot interaction. The following section briefly reviews the aspects of interdependence theory

that are used in this chapter. Next, an algorithm, which uses aspects of interdependence theory to produce information about social situations, is detailed. Afterwards, we develop a complete computational process by which a robot can use perceptual information to guide interactive behavior.

6.1.1 Interdependence theory

Recall that interdependence theory is based on the claim that people adjust their interactive behavior in response to their perception of a social situation's pattern of rewards and costs and represents social situations computationally as an outcome matrix (Figure 6.1). Figure 6.1 shows the outcome matrix for our toxic spill cleanup example. The preceding chapter presented methods for creating outcome matrices from perceptual information such as strings of speech. In this chapter we assume that outcome matrices representing a social situation can be created, and begin to look at the advantages of using outcome matrices as a representation of social interaction.

Kelley and Thibaut conducted a vast analysis of both theoretical and experimental social situations and were able to generate a space that mapped particular social situations to the dimensional characteristics of the situation (Kelley & Thibaut, 1978). Recall that the interdependence space (Figure 6.2 depicts three of the four dimensions) is a four dimensional space consisting of: (1) an interdependence dimension, (2) a correspondence dimension, (3) a control dimension, and (4) a symmetry dimension. The interdependence dimension measures the extent to which each individual's outcomes are influenced by the other individual's actions in a situation. In a low interdependence situation, for example, each individual's outcomes are relatively independent of the other individual's choice of interactive behavior (left side of Figure 6.1 for example). A high interdependence

situation, on the other hand, is a situation in which each individual's outcomes largely depend on the action of the other individual (right side of Figure 6.1 for example). Correspondence describes the extent to which the outcomes of one individual in a situation are consistent with the outcomes of the other individual. If outcomes correspond then individuals tend to select interactive behaviors resulting in mutually rewarding outcomes, such as teammates in a game. If outcomes conflict then individuals tend to select interactive behaviors resulting in mutually costly outcomes, such as opponents in a game.

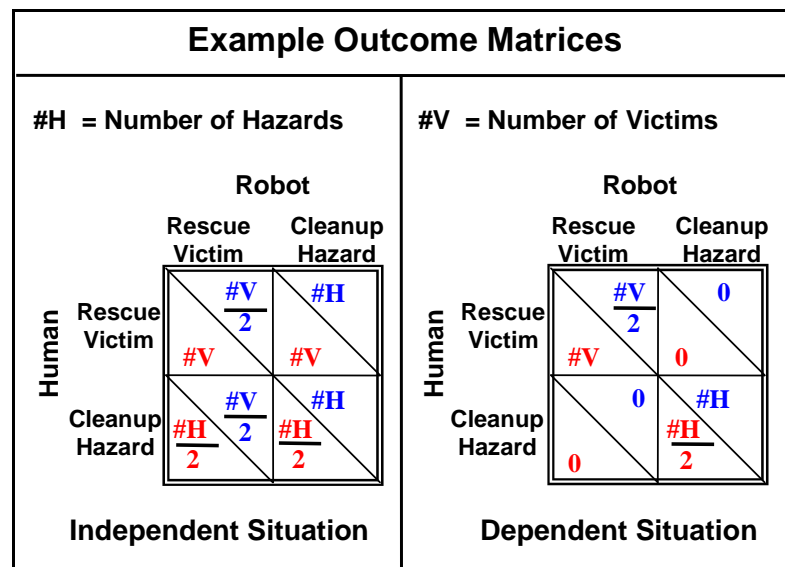


Figure 6.1 This figure depicts two example outcome matrices for the cleanup of a toxic spill and the rescue of victims by a human and a robot. During any one interaction, both individuals choose to either rescue a victim or clean up a hazard. The outcomes resulting from each pair of choices are depicted in the cells of the matrix. The human's outcomes are listed below the robot's outcomes. In the leftmost matrix, the outcomes for the human and the robot are independent of the other's action selection. In the rightmost matrix, the outcomes of the human and the robot largely depend on the other's action selection.

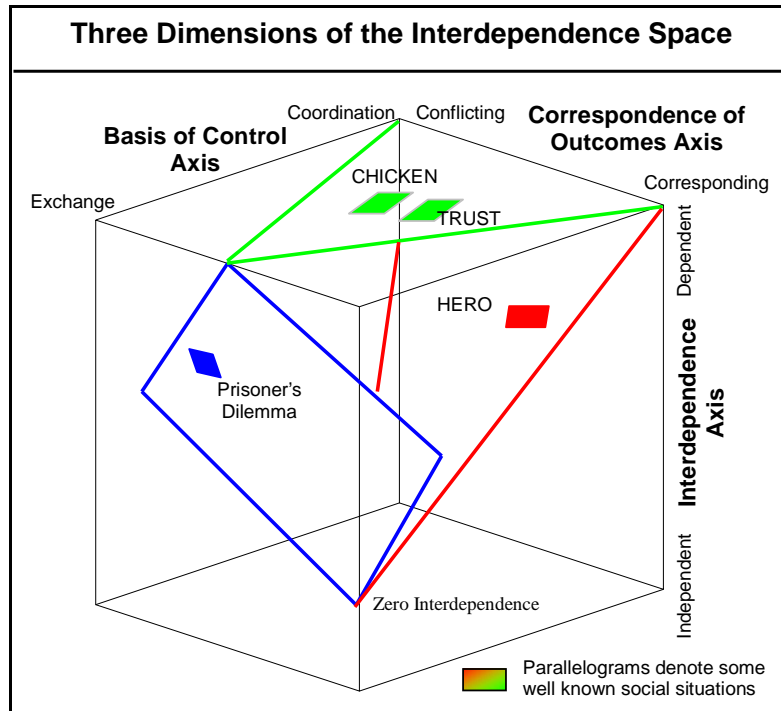


Figure 6.2. Three dimensions of interdependence space are depicted above (Kelley et al., 2003). Interdependence theory represents social situations computationally as an outcome matrix within this interdependence space. The dimensions depicted above are interdependence, correspondence, and basis of control. Planes within this space denote the location of some well-known social situations, including the prisoner’s dilemma game, the trust game, and the hero game. A matrix’s location allows one to predict possible results of interaction within the situation.

A matrix’s location in interdependence space provides important information relating to the situation. For example, in a situation of low interdependence the robot should generally select the behavior that maximizes its own outcome, because its choice of action will not have a large impact on the outcome of its partner. We term the process of deconstructing a matrix into its interdependence space dimensions *situation analysis*. As will be demonstrated, the information provided by situation analysis can be used to profitably guide interactive behavior selection by a robot.

6.1.2 The situation analysis algorithm

Situation analysis is a general technique we developed from interdependence theory to provide a robot with information about its social situation. As an algorithm, it can be used

in an on-line or an off-line manner to provide information about any social situation represented by an outcome matrix. Thus, in theory, a robot could use situation analysis as a tool to investigate potential social situations it might encounter or situations that have occurred in the past among others. The input to the algorithm is an outcome matrix representing the social situation. The algorithm outputs a tuple, $\langle \alpha, \beta, \gamma, \delta \rangle$, indicating the situation's location in the four dimensional interdependence space. Situation analysis involves 1) deconstructing the outcome matrix into values representing the variances in outcome and 2) the generation of the dimensional values for the interdependence space. Box 6.1 describes situation analysis algorithmically.

The Situation Analysis Algorithm

Input: Outcome Matrix O

Output: Interdependence space tuple $\langle \alpha, \beta, \chi, \delta \rangle$

1. Use procedure from Figure 6.3 to deconstruct the outcome matrix.
2. Use the equations from Table 6.1 to calculate the dimensional values for the interdependence space tuple.
3. **Return** the tuple.

Box 6.1 An algorithm for the analysis of a social situation.

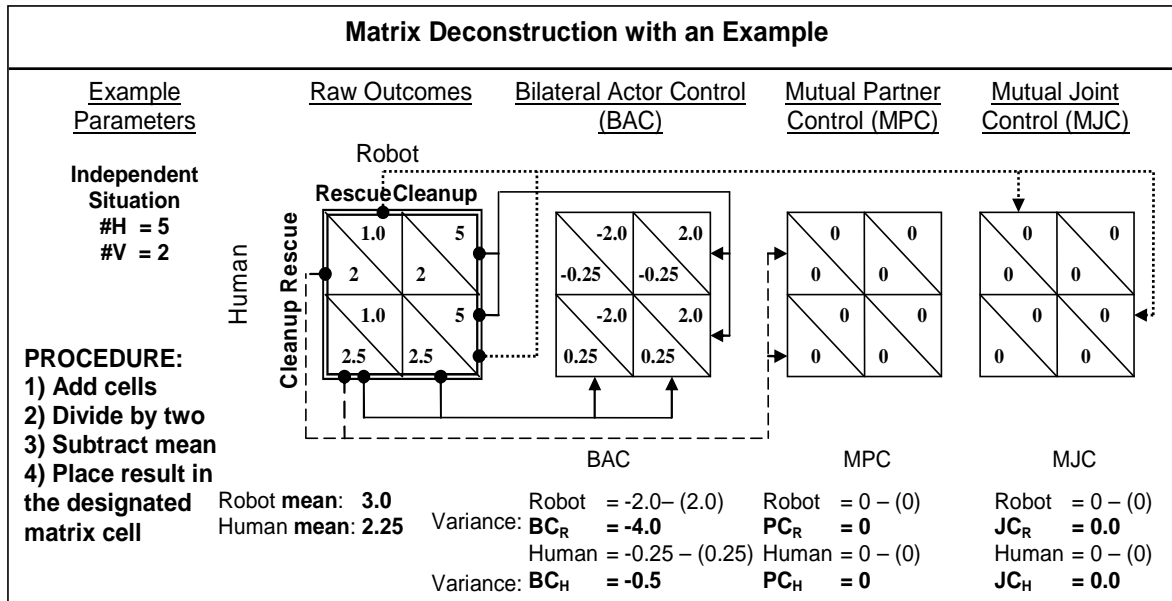


Figure 6.3. The procedure (Kelley & Thibaut, 1978) for deconstructing a social situation is presented above. This procedure is an analysis of variance of the outcome matrix that deconstructs the raw outcome matrix into three new matrices (the BAC, MPC, and MJC) representing different forms of control over the situation's outcomes. The outcome values for each of these three matrices are produced from the raw outcome matrix by iteratively 1) adding the noted cells, 2) dividing by the number of actions, and 3) subtracting the individual's mean outcome value. The variances of each matrix type are generated by calculating the outcome range for each choice of behavior and each individual. Because this example is of an independent situation, the MPC and MJC matrices do not vary.

The first step is matrix deconstruction. This procedure iteratively separates the values in the input or raw outcome matrix into three separate matrices (Figure 6.3 depicts an example) (Kelley & Thibaut, 1978). The Bilateral Actor Control (BAC) matrix represents the variance in outcome resulting from the robot's own interactive decisions. This matrix thus quantifies the robot's control over its own outcomes. The Mutual Partner Control (MPC) matrix, on the other hand, represents the variance in outcome resulting from a partner's interactive decisions and thus quantifies a partner's control over the robot's outcomes. Finally, the Mutual Joint Control (MJC) matrix represents the variance in outcome resulting from both the robot's and its partner's joint interactive decisions. In other words, the MJC matrix describes how each individual is affected by his, her, or its joint actions. As depicted in Figure 6.3, all outcome variance occurs in the BAC matrix when deconstructing an independent situation. This procedure results in values for

variables BC , PC , JC individually representing the variance of both the robot's and the human's outcomes in the situation. The subscripts in this figure denote the variance of the outcome for the robot (R) and the human (H) respectively.

Table 6.1 Calculation of the interdependence space dimensions given the variances from Figure 6.3. Equations (4) and (5) are from (Kelley & Thibaut, 1978), (6) and (7) were developed by the author.

Dimension	Computation
Interdependence (α_R, α_H)	$\alpha_R = \frac{(PC_R^2 + JC_R^2)}{(BC_R^2 + PC_R^2 + JC_R^2)} \quad (4)$ <p>Calculate separately for each individual. Range is from 0 for independent situations to +1 for dependent situations.</p>
Correspondence (β)	$\beta = \frac{2(BC_R PC_H + BC_H PC_R + JC_R JC_H)}{(BC_R^2 + PC_R^2 + JC_R^2 + BC_H^2 + PC_H^2 + JC_H^2)} \quad (5)$ <p>Calculate once for both individuals. Range is from -1 for a situation in which the dyad's outcomes conflict to +1 for a situation in which the dyad's outcomes correspond.</p>
Basis of Control (γ)	$\gamma = \frac{4(\sigma - \nu)}{(Sum(sit))^2} \text{ where} \quad (6)$ $\sigma = (JC_R + JC_H)^2 + (JC_R - JC_H)^2$ $\nu = (BC_R + PC_H)^2 + (BC_H + PC_R)^2 + (BC_R - PC_H)^2 + (BC_H - PC_R)^2$ <p>Calculate once for both individuals. Range is from -1 for a situation controlled by exchange and to +1 for a situation controlled by coordination. <i>Sum(sit)</i> is a cell by cell sum of the matrix.</p>
Symmetry (δ)	$\delta = \frac{(BC_R^2 + PC_H^2 + JC_R^2) - (BC_H^2 + PC_R^2 + JC_H^2)}{(BC_R^2 + PC_R^2 + JC_R^2 + BC_H^2 + PC_H^2 + JC_H^2)} \quad (7)$ <p>Calculate once for both individuals. Range is from -1 for an asymmetric situation in which individual R depends on H to +1 for an asymmetric situation in which individual H depends on R. The value of 0 denotes a symmetric situation (i.e. mutual dependence).</p>

Once the variances for the situation have been computed these values can be used to calculate the situation's location in interdependence space. This is accomplished using equations (4-7) from Table 6.1. Equations (4) and (5) are from (Kelley & Thibaut, 1978). Equations (6) and (7) are contributions of this dissertation. Equation (4) subtracts the outcome resulting from joint action by the individual's from the outcome resulting from partner and individual control. This value is then normalized. Equation (5) subtracts one

individual's control over their own outcomes from the other individual's control. This value is normalized with respect to both individual's outcomes. These values constitute the tuple $\langle \alpha, \beta, \gamma, \delta \rangle$, the situation's location in interdependence space.

6.1.3 Using situation analysis to select interactive behaviors

The situation analysis algorithm presented above begs several questions. Notably, 1) how are the outcome matrices created? 2) How is the location in interdependence space used to control a robot's behavior? 3) Does knowing a situation's location in interdependence afford valuable information for determining which behavior to select? This section addresses each of these questions in turn.

The previous chapter has discussed in detail our methods for creating outcome matrices. For the experiments conducted as part of this research, the number of hazards and victims perceived is used to construct the outcome matrix (Figure 6.1). These matrices expand upon the human-robot cleanup situation described previously. In these examples, both the human and the robot select either an action to rescue a victim or to cleanup a hazard. The outcome for each pair of selected actions, in this case, is a function of the number of victims and hazards in the environment. The functions in Figure 6.1 were selected to give the autonomous robot a preference for cleanups and the teleoperated robot a preference for victims. Preferences such as these might result from the configuration of each robot. In the independent situation, for example, if the robot chooses to cleanup a hazard and the human chooses to rescue a victim, then the human obtains an outcome equal to the number of victims and the robot obtains an outcome equal to the number of hazards. In the dependent condition, on the other hand, positive

outcome is only obtained if both the robot and the human select the same action. A situation such as this could occur if victims must be cleaned prior to being rescued.

Table 6.2 A list of several simple matrix transformations. The list is not exhaustive.

Transformation name	Transformation mechanism	Social character
<i>max_own</i>	No change	Egoism —the individual selects the action that most favors their own outcomes
<i>max_other</i>	Swap partner's outcomes with one's own	Altruism —the individual selects the action that most favors their partner
<i>max_joint</i>	Replace outcomes with the sum of the individual and the partner's outcome	Cooperation —the individual selects the action that most favors both their own and their partner's outcome
<i>max_diff</i>	Replace outcomes with the difference of the individual's outcome to that of the partner	Competition —the individual selects the action that results in the most relative gain to that of its partner
<i>min_diff</i>	Maximize the value of the action that has the minimal difference to that of the partner.	Fairness —the individual selects the action that results in the least disparity
<i>min_risk</i>	Maximize the value of the action that has the greatest minimal outcome	Risk-aversion —the individual selects actions that result in the maximal guaranteed outcomes

Before discussing how this information is used to control a robot's behavior, we consider strategies by which the outcome matrix can be directly used to select actions. The most obvious method for selecting an action from an outcome matrix is to simply choose the action that maximizes the robot's outcome. We term this strategy *max_own*. Alternatively, the outcome matrix can be transformed to create a new, different matrix that the robot uses to select a behavior. Table 6.2 lists several different methods for transforming an outcome matrix. In the case of *max_other* the partner's outcome values are swapped with the robot's outcome values. The *max_joint* transformation, on the other hand, replaces the robot's outcomes with the sum of the robot and its partner's outcome. Once an outcome matrix has been transformed, the *max_own* strategy is used to select an action. This simple technique of transforming the outcome matrix and then using the *max_own* strategy to select a behavior serves as a control strategy and has the benefit of

changing the character of the robot's response without consideration of the actual actions involved.

Because the situation analysis algorithm simply provides information, this information could theoretically be used in many different ways to aid action selection. For instance, rules could directly map a situation's location to a particular action. Alternatively, the information could be used to select transformations (Table 6.2). One advantage of the latter method is that it does not require knowledge of the actions available to the robot. Rather, the situation's interdependence space location is used to alter the robot's response *independent of interactive actions available*. Another advantage of this approach is that one can test a specified set of transformations at a given location to determine which transformation is best at that location. In this manner, a mapping of interdependence space location to transformation can be developed which is independent of the individuals interacting and the actions available. As will be discussed in the next section, our initial step for this research was creating this mapping of situation location to transformation.

Finally, does knowing a situation's location in interdependence space afford valuable information? We approached this question empirically by performing two experiments in simulation. The first experiment investigates the value of this information in a practical scenario. The second experiment considers the value of knowing the situation's location over the entire interdependence space.

6.1.4 Mapping a situation's location to a transformation

A mapping from a situation's location to a transformation can be described formally as the function $f : L \rightarrow T$ where L is the interdependence space location and T is the space

of possible transformations. We subdivide the interdependence space into three areas of interest to robotics researchers, namely high interdependence ($\alpha_R \geq 0.75$) and low correspondence ($\beta \leq 0$), high interdependence ($\alpha_R \geq 0.75$) and high correspondence ($\beta > 0$) and low interdependence if $\alpha_R < 0.75$. These areas are abbreviated as l_{hl}, l_{hh}, l_l respectively. The area l_{hl} represents situations in which the robot's outcomes greatly depend on its partner but the robot and the human do not have action preferences towards the same goal, potentially resulting in poor outcomes for the robot. The area l_{hh} , on the other hand, describes situations in which the robot's outcomes also greatly depend on its partner and both the robot and the human have action preferences towards the same goal. Finally, the area l_l represents the location of situations in which the robot's outcomes do not greatly depend on its partner. Thus $L = \{l_{hl}, l_{hh}, l_l\}$ describes the domain of f . The codomain of f is the set of transformations considered as part of this work (see Table 6.2 for descriptions).

Given the preceding description, the challenge then is to determine for each location in L which transformation from T results in the greatest overall net outcome. To do this we created a random matrix and then used the situation analysis algorithm to determine the matrix's location in interdependence space until we had 1000 matrices in each area l_{hl}, l_{hh}, l_l . Random matrices consisted of an empty matrix populated with random numbers between 0 and 24. The number 24 was arbitrarily selected. Next, for every matrix in each area l_{hl}, l_{hh}, l_l , we iterated through the set T altering the matrix according to the transformation's specification (Table 6.2). Afterward, a simulated robot selects the action from the transformed matrix that maximizes its outcome. The robot's simulated

partner also selects an action from the original matrix that maximizes its outcome. Finally, the robot’s outcome resulting from the action pair (as dictated by the original matrix) is recorded. Figure 6.5 in section 6.2.1 graphically depicts this procedure and the other experimental procedures used.

Table 6.3 The cells denote the mean outcome obtained by the transformation at each location. The shaded cells indicate the mean of the best transformation. The confidence interval is included for all values.

Low interdependence		High interdependence/high correspondence		High interdependence/low correspondence	
Transformation	Mean outcome	Transformation	Mean outcome	Transformation	Mean outcome
<i>max_own</i>	13.47 ± 0.46	<i>max_own</i>	15.01 ± 0.39	<i>max_own</i>	14.27 ± 0.41
<i>min_own</i>	10.36 ± 0.46	<i>min_own</i>	8.75 ± 0.40	<i>min_own</i>	7.712 ± 0.38
<i>max_other</i>	11.67 ± 0.43	<i>max_other</i>	15.10 ± 0.36	<i>max_other</i>	7.80 ± 0.37
<i>min_other</i>	11.86 ± 0.43	<i>min_other</i>	10.52 ± 0.42	<i>min_other</i>	12.94 ± 0.42
<i>max_joint</i>	12.90 ± 0.43	<i>max_joint</i>	16.03 ± 0.34	<i>max_joint</i>	13.40 ± 0.42
<i>min_joint</i>	11.16 ± 0.44	<i>min_joint</i>	9.55 ± 0.41	<i>min_joint</i>	10.52 ± 0.43
<i>max_diff</i>	11.41 ± 0.46	<i>max_diff</i>	10.41 ± 0.43	<i>max_diff</i>	9.93 ± 0.47
<i>min_diff</i>	12.08 ± 0.42	<i>min_diff</i>	12.48 ± 0.43	<i>min_diff</i>	12.10 ± 0.41
<i>min_risk</i>	13.08 ± 0.41	<i>min_risk</i>	14.82 ± 0.38	<i>min_risk</i>	14.79 ± 0.37

Table 6.3 presents the mean outcome resulting from each transformation at each location. The transformation that results in the greatest mean outcome for each location in shaded. Note that the difference in mean outcome for several of the transformations is not great. This lack of difference reflects the similarity of the transform in the particular area of interdependence space. More importantly, it foreshadows the need of a robot to interact with its partner in a variety of situations located at different positions in interdependence space in order to determine the partner’s transformation preference or type. The table indicates that *max_own*, *max_joint*, and *min_risk* are the best transformations of the group of possible transformations in low interdependence, high interdependence/high correspondence, and high interdependence/low correspondence situations respectively. From this data the function f mapping the interdependence space

location to transformation takes the following form, $f(l^*) = \begin{cases} \text{max_own} & l^* = l_l \\ \text{max_joint} & \text{for } l^* = l_{hh} \\ \text{min_risk} & l^* = l_{hl} \end{cases}$

where l^* is the interdependence space location generated by the situation analysis algorithm. This function can also be visualized as the decision tree in Figure 6.4.

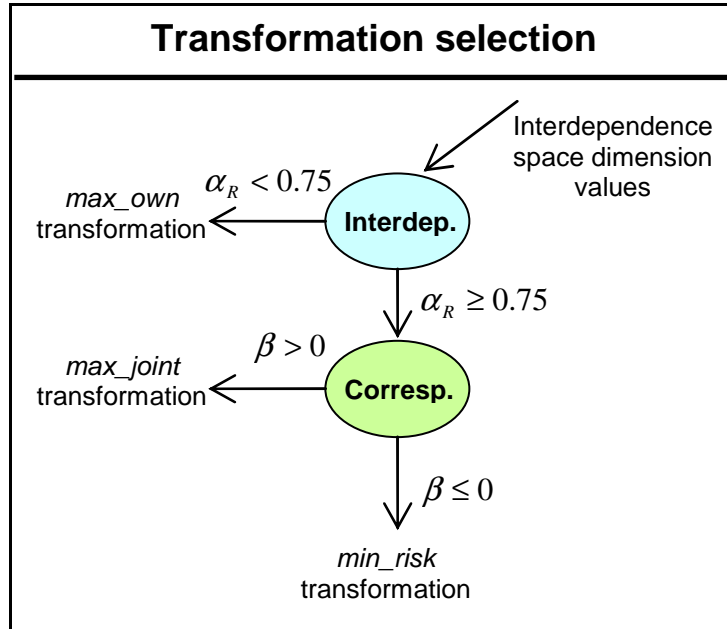


Figure 6.4 A mapping of interdependence space location to outcome matrix transformation.

We have therefore found a mapping from a situation’s location in interdependence to a transformation. This mapping allows us to create a computational process that begins with the outcome matrix and will end with the selection of an action. The next section develops the remainder of this process.

6.1.5 A computational process for situation analysis

Assuming that outcome matrices can be generated and given the mapping from interdependence location to transformation developed in the preceding section, a computational process can be developed that selects a robot’s behavior from its perception of the situation. This computational process is depicted in Figure 6.5. The

right side of this figure depicts a stepwise procedure for generating interactive action from perception. The first step is the creation of an outcome matrix. In our experiments, these were either derived perceptually, by recognizing objects in the environment and using the matrices in Figure 6.1, or generated by populating an empty matrix with random values. The next two steps consist of the situation analysis algorithm described in section 6.1.2, which results in an interdependence space tuple. This tuple is then mapped to a transformation using the function f (also depicted in Figure 6.4). The transformation is used to transform the original matrix in the next step. The transformation process results in the construction of an outcome matrix on which the robot can act—the effective situation (Kelley & Thibaut, 1978). In the final step, the robot selects the action in the effective situation that maximizes its own outcome. The left side of Figure 6.5 depicts an example run through the procedure. The next section discusses our empirical examination of this process.

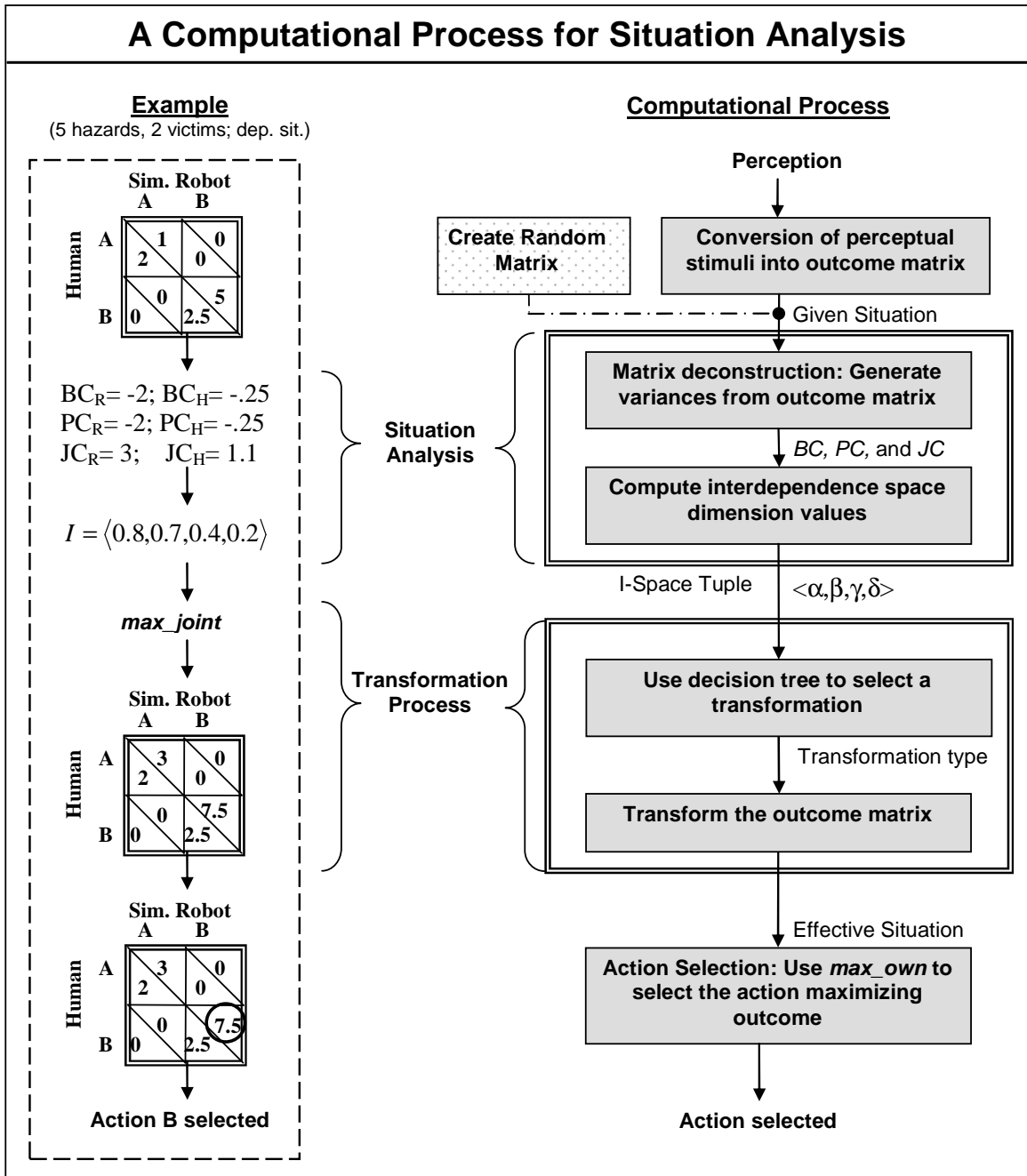


Figure 6.5 This figure depicts the algorithmic process contributed by this work. The process consists of six steps. The first step generates an outcome matrix. The second step analyzes the matrix's variances. The third step computes the situation's interdependence space dimensions. These two steps constitute the process of situation analysis. The fourth step selects a transformation and in the fifth step, the transformation is applied to the outcome matrix resulting in the effective situation. Steps 4 and 5 constitute the transformation process. Finally, an action is selected.

6.2 Experiments and Results

The preceding discussion has described *how* an outcome matrix can be mapped to a location in interdependence space and *how* information about the matrix's location can be used to select a robot's interactive action. We have not yet shown, however, that the information afforded by the situation analysis algorithm results in better interactive behavior on the part of the robot. The experiments presented in this section, therefore, examine the value of the information generated by the situation analysis algorithm. Value here is operationalized as increase in net outcome. Both experiments test the hypothesis that the use of the situation analysis algorithm will result in an increase in net outcome when compared to alternative control strategies. The first experiment uses the computational process from Figure 6.5 to guide a simulated robot's action selection in the cleanup and rescue example described at the beginning of the chapter. The second experiment generalizes the results from the first experiment to the entire interdependence space and compares the algorithm to a larger number of control strategies.

6.2.1 Situation analysis in practice

To revisit the scenario described at the beginning of the chapter, a teleoperated robot attempts to rescue victims of an industrial accident while an autonomous robot works to cleanup a spill. We considered two scenarios in simulation: one involving greater dependence (high interdependence condition) and another involving little dependence (low interdependence condition). Notionally, because of the properties of the chemical the high interdependence condition requires that the victims be cleaned before being rescued. Thus, in this condition, the robots must both cooperate in order to complete the

rescue task successfully. In the low interdependence condition, both robots can operate independently of one another. This scenario is based on the well-studied foraging problem in robotics (Arkin, 1999). Figure 6.6 depicts the layout. Potential victims and hazards for cleanup are located within a disaster area. A disposal area for hazardous items is located towards the bottom and a triage area for victims is located to the right. Table 6.4 summarizes the experiment.

Table 6.4 Experimental summary for the situation analysis experiment conducted in a search and rescue environment.

Experiment Summary	
Situation Analysis in Practice	
Purpose	Explore the use of information pertaining to a situation's position in the interdependence space to control a robot's behavior.
Experiment Type	MissionLab simulation environment
Hypothesis	That the use of the situation analysis algorithm results in an increase in net outcome when compared to alternative control strategies in dependent situations.
Procedure	Follow partner model creation procedure from Figure 6.7.
Independent variable	Conditions: use of situation analysis information versus no use of situation analysis information; dependent versus independent situation.
Dependent variable	Net outcome
Method of Analysis	Ablation experiment consisting of comparison between use of situation analysis information and no use of situation analysis information.
Conclusion	Hypothesis is supported. Conditions in which the situation was dependent and the situation analysis algorithm was used resulted in greater outcome being obtained compared to not using the situation analysis algorithm.

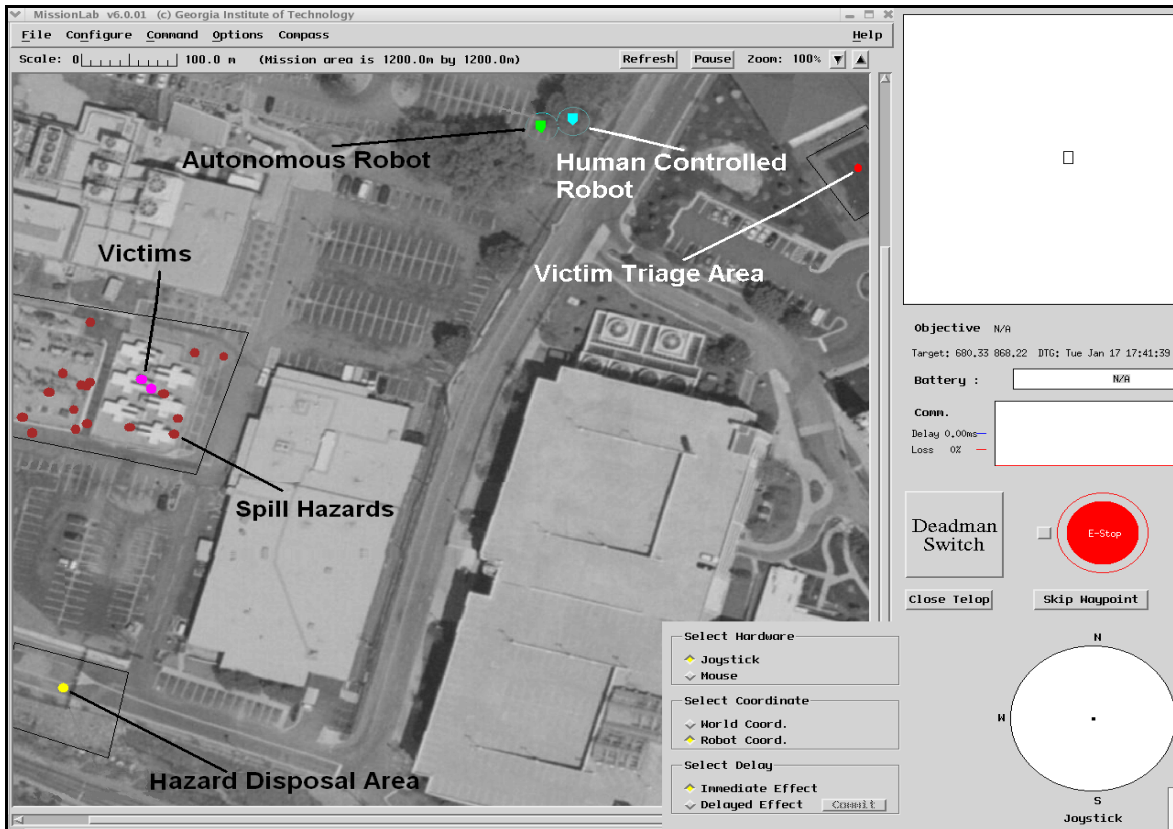


Figure 6.6 The simulation environment used for the cleanup and rescue experiment is depicted above. The experiment required that a teleoperated robot rescue victims while an autonomous robot performs a cleanup. Experimental conditions included independent versus dependent situations and the use of our situation analysis algorithm versus a control strategy. The teleoperation interface used by the human is depicted the right.

This experiment compares the net outcome obtained by both robots as well as the number of victims rescued and hazards cleaned in four separate conditions. In the experimental conditions, the autonomous robot used the computational process depicted in Figure 6.5 to select its action. In the control conditions, the autonomous robot consistently selected the behavior that maximized its own outcome without consideration of its partner (*max_own*). The experimental and control condition were explored in both high interdependence situations and low interdependence situations. A high interdependence situation was created by populating the dependent outcome matrix from Figure 6.1. Similarly, a low interdependence situation was created by populating the independent outcome matrix from the Figure 6.1. Thus, the experiment consisted of the

following four conditions: high interdependence-situation analysis, high interdependence-control strategy, low interdependence-situation analysis, low interdependence-control strategy. In all conditions, the teleoperated robot selected the behavior that maximized its own outcome without consideration of its partner (*max_own*). The primary author controlled the teleoperated robot. Because the teleoperated robot employs a static actor script, experimenter bias is eliminated.

Figure 6.7 describes the experimental procedure used (middle procedure). First, a random number of victims and hazards were generated. Next, a Gaussian distribution was used to randomly place the victims and hazards in the environment. In the low interdependence condition, the autonomous robot perceives the number of victims and hazards and uses the independent matrix from Figure 6.1 to create its outcome matrix. In the high interdependence condition, the autonomous robot uses the dependent matrix to create its outcome matrix. The outcome matrix is then tested using the situation analysis algorithm and the control strategy. The behaviors that the robot selects are actually collections of actions that direct the robot to locate the closest attractor, pickup the attractor, transport the attractor to a disposal area where it is dropped off and finally return to a staging area. The *MissionLab* mission specification system was used. *MissionLab* is a graphical software toolset that allows users to generate mobile robot behavior, test behaviors in simulation, and execute

collections of behaviors on real, embodied robots (MacKenzie, Arkin, & Cameroon, 1997).

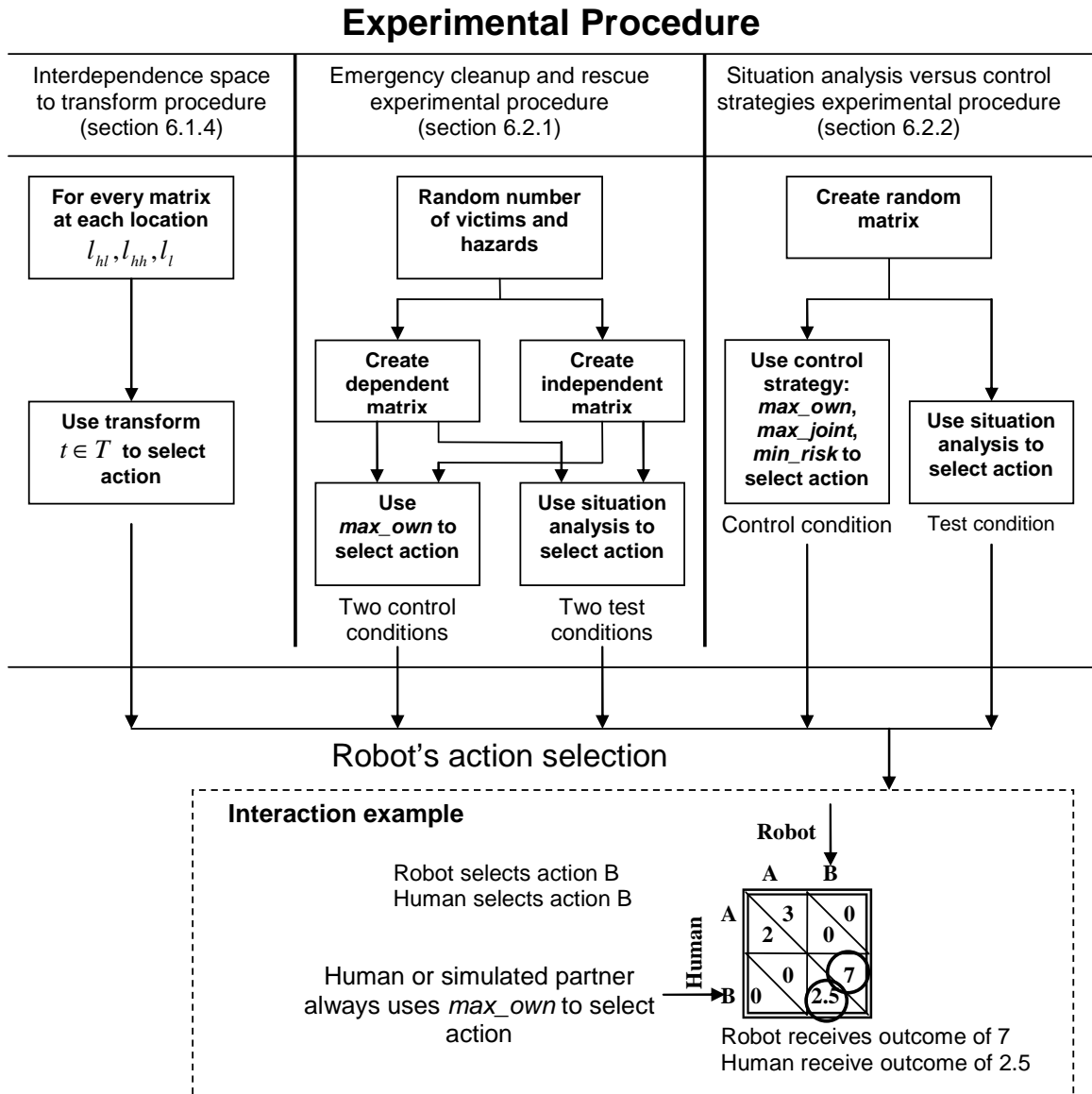


Figure 6.7 The procedures used to create and use outcome matrices are depicted above. The left side details the procedure used to generate Table 6.3. This procedure first iterates through all matrices in each areas l_{hl}, l_{hh}, l_l and then iterates through the set of transformations to produce the matrix the robot will use to select actions. The middle procedure first creates a random number of victims and hazards. Next, an independent and dependent matrix is created from the number of victims and hazards. Finally, in the control conditions, *max_own* is used to select an action. In the test procedure, situation analysis is used to select an action. The right most procedure, first generates a random matrix and then transforms the matrix with respect to a control matrix or uses situation analysis. The robot selects an action from the transformed matrix. The interaction example at the bottom denotes the method used to determine how much outcome each individual receives from the presentation of an outcome matrix.

We conducted thirty trials in each of the four conditions: independent situation/control robot, independent situation/test robot, dependent situation/control robot, dependent situation/test robot. In these experiments, interaction occurs when both individuals (autonomous robot and teleoperated robot, or both simulated robots) are presented with an outcome matrix and simultaneously select actions from the matrix receiving the outcome that results from the action pair. We recorded the number of victims rescued and the hazards collected after each trial. We predicted that the situation analysis algorithm would outperform the control strategy in the dependent condition but not in the independent condition. Independent situations, by definition, demand little consideration of the partner's actions. Thus, in these situations, the autonomous robot's performance is not affected by the actions of the partner. Dependent situations, on the other hand, demand consideration of the partner, and we believed that our algorithm would aid performance in these conditions. The procedure tests the hypothesis by comparing task performance (number of victims and hazards retrieved) with and without the situation analysis information.

Figure 6.8 illustrates the results from the cleanup and rescue experiment. The left two bars portray the results for the independent situation. In these conditions, the autonomous robot forages for hazards to cleanup and the human-operated robot uses MissionLab's search and collect behaviors to forage for victims. Thus, in all of the 30 trials each robot retrieves either a victim or a hazard. As predicted, the robot using situation analysis information and the robot not using situation analysis information both retrieve 30 victims and 30 hazards in this condition.

Cleanup and Rescue Experiment Results

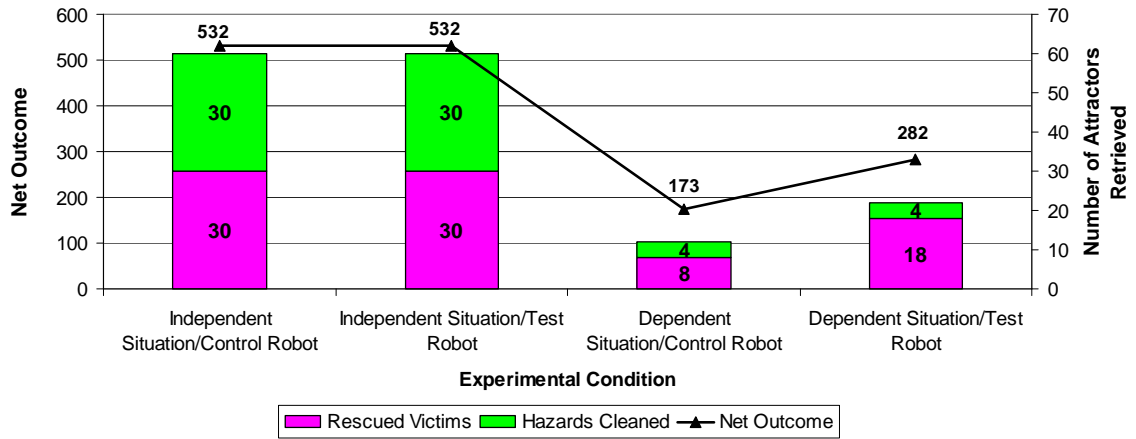


Figure 6.8 Results for the cleanup and rescue experiment are presented above. The line graph portrays the net outcome for each condition. The bars depict the number of hazards and victims retrieved. Hazards cleaned are shown above the number of victims rescued. The left two bars and line points depict the independent conditions for both the test and the control robot. In these conditions both the control and test robot perform equally well. The right two bars and line points examine the dependent situation. Note that in this situation the test robot outperforms the control robot.

In the dependent condition, because the retrieval of a victim or a hazard required the cooperation of both robots, the best possible score was thirty. The autonomous robot’s use of situation information results in ten additional victims being rescued. Thus, as predicted, in the dependent condition the autonomous robot’s use of situation information affords better performance than the robot that does not consider the situation. In this case, the information provided by our algorithm indicates to the autonomous robot that its outcomes for this situation rely on collaboration with its human-operated partner. The control strategy, on the other hand, fails to consider the partner’s role even though the situation demands collaboration, hence resulting in poorer performance.

Overall, this experiment demonstrates that the information resulting from an analysis of the social situation can improve a robot’s ability to perform interactive tasks similar to collaborative foraging. The algorithm we have proposed uses perceptual stimuli in the environment to produce information about the social situation. Minimally, we have

shown the feasibility of our approach and the potential importance of situational considerations in human-robot interaction, ideas which have not been investigated as a part of HRI in the past. Nevertheless, the results of this experiment are limited in several ways. First, the situations encountered as part of the experiment are derived from a limited portion of the interdependence space. Second, only a single control strategy was considered. The next experiment generalizes these results to the entire interdependence space and considers additional controls.

6.2.2 Situation analysis over the entire interdependence space

Whereas the previous experiment only explored high interdependence or low interdependence outcome matrices, this experiment considers outcome matrices from every corner of the interdependence space. We examine the algorithm's performance over thousands of different matrices representing a broad spectrum of the interdependence space. Because of time-constraints, it was not possible to test each of these matrices using interaction between a human and a robot. Rather, the human was replaced with an agent that selected the behavior that maximized its own outcome without consideration of its partner (*max_own*). The same strategy was employed by the human in the first experiment and the agent in this experiment.

For this experiment, we also compare the algorithm's performance to four different control strategies. For the first control strategy, the autonomous robot consistently selected the behavior that maximized its own outcome without consideration of its partner (*max_own*). For the second control strategy, the autonomous robot consistently selected the behavior that minimized the difference of its and its partner's outcome (*min_diff*). For the third control strategy, the autonomous robot consistently selected the

behavior that maximizes the sum of its and its partner's outcome (*max_joint*). For the final control strategy, the autonomous robot consistently selected the behavior that resulted in the greatest guaranteed outcome (*min_risk*).

Table 6.5 Experimental summary for the situation analysis experiment conducted over the entire interdependence space.

Experiment Summary	
Situation Analysis over the entire Interdependence Space	
Purpose	Explore the use of information pertaining to a situation's position in the interdependence space to control a robot's behavior.
Experiment Type	Numerical simulation
Hypothesis	That the use of the situation analysis algorithm results in an increase in net outcome when compared to alternative control strategies.
Procedure	Follow partner model creation procedure from Figure 6.7.
Independent variable	Action selection strategy.
Dependent variable	Net outcome
Method of Analysis	Comparison of several different alternative action selection strategies to the use of situation information.
Conclusion	Hypothesis is supported. The use of situation analysis information results in significantly greater net outcome being obtained by the robot than does any of the control strategies.

Figure 6.7 describes the experimental procedure used (right procedure). First, a random matrix is created from an empty matrix populated with random numbers between 0 and 24. The random matrix in this case does not have actions assigned. Hence, these matrices are abstract in the sense that the rewards and costs are associated with selecting one of two non-specified actions. Once a matrix is created, it is presented to both the simulated robot and the agent. Both simultaneously select actions from the matrix receiving the outcome that results from the action pair. The simulated robot uses either situation analysis or one of the previously discussed control strategies (section 6.2.1) to determine which action to select from the matrix. This experiment was conducted as a numerical simulation and hence did not occur in a robot simulation environment. In other words, the simulated robot in this case was an agent that selects an action in accordance with the strategy dictated by the experimental condition, but did not actually have to

perform the action in an environment. Consequentially, this experiment did not require perceptual generation of the outcome matrix and the actions selected by the agents did not affect the environment. One consequence is that the conclusions drawn from the results of this experiment do not relate to a robot or task in particular.

In order to ensure coverage over the entire space, we examined one hundred trials each consisting of 1000 randomly generated outcome matrices. We recorded the outcome obtained by each individual for the pair of actions selected. We predicted that the net outcome received by the simulated autonomous robot would be greater and statistically significant when the robot used the computational process from Figure 6.5 when compared to the controls. We reasoned that, on average, the information provided by situation analysis would be valuable to the robot for its selection of its behavior. We thus hypothesized that the use of this information would result in a greater net outcome than the control strategies.

Figure 6.9 presents results for this experiment. The second bar from the left depicts the net outcome using the situation analysis algorithm. The next four bars to the right indicate the net outcome for the control conditions. Our algorithm significantly outperforms the controls in all four conditions ($p < 0.01$ two-tailed, for all). The maximum possible outcome for a robot with complete a priori knowledge of all of its partner's actions is also depicted to the left for reference.

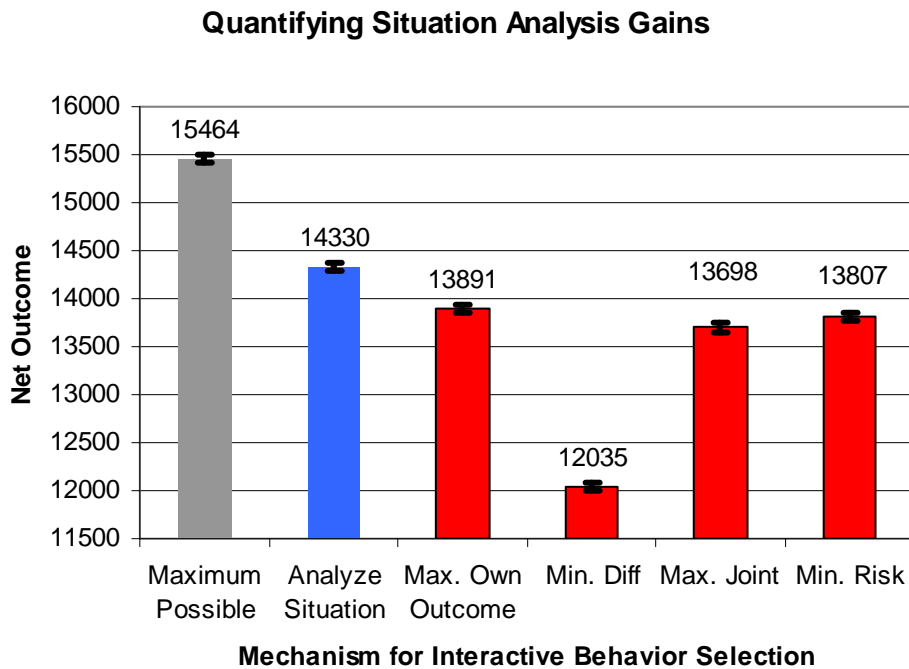


Figure 6.9 Results of this second experiment are presented above. The second bar from the left indicates the net outcome when the situation analysis algorithm is used. The next four bars are the controls for the experiment. Error bars indicate 95% confidence interval. Analyzing the situation resulted in the greatest net outcome of when compared to the control strategies. The leftmost bar portrays the maximum possible net outcome. Note that use of the situation analysis algorithm results in significantly greater outcome than the other control strategies.

The results confirm our prediction that use of the situation analysis algorithm results in greater net outcome than does the use of the control strategies. The graph also indicates that our procedure outperforms the four different control strategies. Furthermore, the results show that our procedure is beneficial on average to an agent or robot that will face many different social situations from unique locations in the interdependence space. Still, the algorithm performs far below the maximum possible. Better performance could likely be achieved by increasing the size of the domain and codomain of f , the mapping from interdependence space location to transform (from section 6.1.4). In this work, we subdivided the interdependence space into three areas, denoted l_{hl}, l_{hh}, l_l . Greater subdivision of the space would make better use of the information provided by the

situation analysis algorithm. We also limited the number of transformations considered to nine. Additional transformations would increase the algorithm's performance if a novel transformation outperformed all other transformations at some location in the space.

The value of the situation analysis algorithm, as presented in this chapter, stems from the very fact that it knows nothing of its interactive partner. The computational process does not assume anything about the partner. Rather it operates only on the information available within the outcome matrix. This is in contrast to game theory, which operates on the presumption of the partner's rationality (Osborne & Rubinstein, 1994). We expect that the performance of this approach would increase drastically as additional, partner specific, information is provided.

6.3 Situation Analysis Conclusions

This chapter has introduced a method for capturing information about social situations and for using this information to guide a simulated robot's interactive behavior. We have presented an algorithm for situation analysis and a computational process for using the algorithm. Our approach is derived from the social psychological theory of interdependence and has close ties to the psychology of human-human interaction (Kelley & Thibaut, 1978). The value of knowing a situation's location in interdependence space has been highlighted with experiments indicating that, on average, this information can aid in selecting interactive actions and that in some situations this information is critical for successful interaction and task performance.

We do not address the challenge of managing uncertainty in this chapter. Much work has already addressed this topic with respect to the outcome matrix (Osborne &

Rubinstein, 1994). The uncertainty present in the outcome matrix will result in similar uncertainty in the situation's location in interdependence space.

We have presented one method for using information about a situation's location to guide behavior selection. Our method relates the matrix's location to a transformation of the matrix. For the most part, we have not used all of the information available. We did not, for example, explore the effect of a situation's symmetry on the behavior of the robot. Symmetry describes the balance of control that the robot or its partner has over the other. The value of this dimension could play an important role in determining behavior. This possibility could be explored as part of future research. Moreover, we have assumed throughout that the partner consistently selects the *max_own* transformation. The exploration of different partner types will also be the fruits of future research.

In summary, it is our contention that this approach offers a general, principled means for both analyzing and reasoning about the social situations faced by a robot. Because the approach is general, we believe that it can be applied to a wide variety of different robot problems and domains. The development of theoretical frameworks that include situation-specific information is an important area of study if robots are expected to move out of the laboratory and into one's home. Moreover, because this work is based on research which has already been validated for interpersonal interaction, we believe that it may eventually allow an artificial system to reason about the situation-specific sources of a human's social behavior. The results of this chapter have shown that our theoretical framework, and the representations included therein, can have a strong positive impact on a robot's ability to select actions. Moreover, these results serve as partial evidence towards the second subsidiary question posed in the first chapter—what effect will

deliberation with respect to the social situation have on the robot's ability to select actions? The chapters that follow explore the use of this framework with respect to relationships and then to trust.

CHAPTER 7

REPRESENTING AND REASONING ABOUT RELATIONSHIPS

Relationships are a fundamental aspect of human socialization (Duck, Acitelli, Manke, & West, 2000). Every human being alive today has, from birth, relied on a vast network of other human beings for survival. From the doctor in the delivery room to an army of teachers, instructors, and friends, humans are shaped and guided by their relationships. It is telling that a lack of relationships is one important indicator of social dysfunction (Farrington, 1993). Clearly then socialization is critical for human development as reports of children raised with minimal socialization often indicate severe disorders (Toth, Halasz, Mikics, Barsy, & Haller, 2008). Hence, for humans, having relationships is essential for survival.

Relationships are also critical for learning. Teachers build relationships with their students that are mutually rewarding and often the material taught is specifically tailored for the student (Trigwell, Prosser, & Waterhouse, 1999). Young non-human primates, for example, predicate their learning with respect their relationships—accepting the tuition of only those individuals with which they have strong relationships (V. Horner, personal communication, February 9, 2006).

Relationships are important for cooperation. There are simply some things which cannot be completed successfully without the help of others. Games, such as soccer for instance, require the participation of others. Relationships allow an individual to better predict and reason about the actions of the other person or people in a cooperative or

competitive team. If a robot is to be a good teammate or a good competitor it must build relationships which allow it to predict the actions of the other members of the team.

Finally, relationships impact communication in important ways. Interpersonal communication is often regulated by the characteristics of the relationship (Sears, Peplau, & Taylor, 1991). Speaking to a superior influences not just what is communicated but also how it is communicated (Schatzman & Strauss, 1955). Similarly, how something is communicated often identifies important characteristics of the relationships. For all of these reasons it will be important for a robot to reason about its relationships.

The purpose of this chapter is to begin to develop the theoretical and algorithmic underpinnings that will allow a robot to reason about its relationships. The chapter begins with a definition of the term relationship and uses the framework set forth in the preceding chapters to create methods that allow the robot characterize its relations.

7.1 Relational Disposition

Relationships develop and are defined by the interactions that compose them (Kelley et al., 2003). Interdependence theory describes a relationship between two individuals as a type of summary of the dyad's interactions over a series of interactions. The definition offered by the American Heritage Dictionary concurs. It states that a relationship is “a particular type of connection existing between individuals related to or having dealings with each other (Relationship, 2000).” Both descriptions of the term agree that a relationship represents a distinctive connection between individuals which develops from their having repeated interactions with one another.

Recall from section 4.1 that the selection of actions by both individuals in an interaction results in outcomes for both individual. Using the notational tools developed

in the previous chapters, we can represent **the result** of a series of interactions between two individuals as $[o_0^i, \dots, o_N^i; o_0^{-i}, \dots, o_N^{-i}]$ where N is the number of interactions. Each of the N interactions occurs when both individuals selects an action (a_k^i, a_k^{-i}) and results in an outcome for both individuals (o_k^i, o_k^{-i}) . As will be shown, the pattern of outcomes that results from a series of interactions can be used to describe the overall interdependence properties of these interactions. These interdependence properties characterize the distinctive connection that has developed from the series of interactions. Consider the interactions of teammates. Teammates select actions that result in mutually positive or mutually negative outcomes. In soccer, for example, when a teammate scores a goal, that individual's actions result in positive outcomes for the entire team. Hence, the pattern of outcomes is correspondent. In contrast, the interactions of opponents are typically conflicting, with the actions selected by one individual resulting in contrasting outcomes compared to that of the opponent. Again considering a soccer example, when an opponent scores a goal, positive outcomes result for the opponent while negative outcomes result for one's own team. Hence, we can use the pattern of outcomes to characterize a relationship in terms of its interdependence properties.

Kelley's Two-dimensional Distribution of Relationships

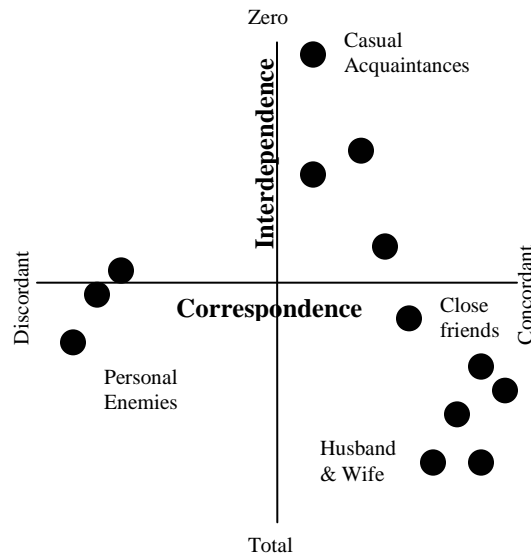


Figure 7.1 Kelley and Thibaut noted that relationships can also be presented within the interdependence space (Kelley & Thibaut, 1978). This figure presents their original mapping of relationships within the interdependence space. Kelley and Thibaut recognized that relationships can be described in terms of interdependence and correspondence, two of the same dimensions that are used to describe social situation.

Kelley and Thibaut note that relationships, like the interactions they accrete from, can be described in terms of their interdependence space location (Figure 7.1). Close relationships, such as that of a husband and wife, tend to be characterized by a high degree of interdependence. Thus the actions of the husband tend to have a large impact on the outcomes of the wife, and vice versa. The interactions of casual acquaintances, in contrast, are marked by little or no interdependence between the two individuals. Correspondence, the extent that each partner's outcomes are consistent with the other partner's outcomes, can similarly be used to map the difference between friends and enemies, with friends having correspondent outcomes and enemies conflicting outcomes. Colloquially the term relationship is often used to describe a particular type of relationship such as mother-daughter, husband-wife, or friends. These and many other relationship types represent generic labels for common interpersonal relationships. Still

they can and have been described with respect to their location in interdependence space (Kelley & Thibaut, 1978). Figure 7.1 depicts the result of research conducted by Kelley and Thibaut relating a dyad's interactions to its two dimensional interdependence space location.

The outcomes $[o_0^i, \dots, o_N^i; o_0^{-i}, \dots, o_N^{-i}]$ represent the end result of several interactions between two individuals. As a roboticist, it is important to develop robots capable of characterizing their developing relationship with a human. To do that, the robot needs to not only recognize the pattern of outcomes that has transpired between it and the human, but must also be able to map that pattern of outcomes back to the human's transformation tendencies. Recall from section 4.3, that an individual transforms the given matrix (O_G) to produce an effective matrix (O_E) which includes the individual's **relational disposition**. Disposition is defined as a stable, social character manifested in an individual. An individual's disposition describes a durative or predominant tendency with respect to an individual's social character. A relational disposition then describes a durative tendency with respect to an individual's relationship with another individual.

Dispositions are exacted via transformation tendencies. Enemies, for example, will tend to have a relational disposition marked by conflict, often attempting to minimize their interactive partner's outcomes. Recall that the transformation process is described formally as $O_E = f(O_G, \theta)$ where O_E is the effective outcome matrix, O_G is the given outcome matrix, θ is the transformation, and the function f transforms the matrix. Interdependence theory originally developed the transformation process from data describing human interaction. Hence, we expect that the transformation process of the robot's human partner can be expressed formally as $O_E^{-i} = f(O_G^{-i}, \theta^{-i})$. Disposition then

is an individual's tendency to select a particular type of transformation. One's relational disposition reflects an individual's tendency to select a particular type of transformation when interacting with a particular individual. Generalization over classes of individuals may also be possible using the stereotype techniques described in section 5.4. Table 7.1 lists several transformations. Consider again, for example, the difference between friends and enemies. Friends will tend to choose prosocial transformations such as *max_joint*, *max_other*, or *min_diff* whereas enemies will tend to choose antisocial transformations such as *min_joint*, *max_own*, *min_other*, and *max_diff*. One problem for the robot then is to determine its partner's relational disposition from a series of interactions with that partner.

Table 7.1 A list of several different types of transformations and a description of each. A relational disposition describes an individual's tendency to use a single transformation when interacting with a particular partner. Hence, the table below describes several relational dispositions.

Relational Disposition Types	
Name	Character Description
<i>max_own</i>	Egoistic —the individual selects the action that most favors their own outcomes.
<i>min_own</i>	Ascetic —the individual selects the action that minimizes his/her own outcomes.
<i>max_other</i>	Altruistic —the individual selects the action that most favors their partner.
<i>min_other</i>	Malevolence —the individual selects the action that least favors the partner.
<i>max_joint</i>	Cooperative —the individual selects the action that most favors both their own and their partner's interests.
<i>min_joint</i>	Vengefulness —the individual selects the action that is most mutually disagreeable.
<i>max_diff</i>	Competitive —the individual selects the action that results in the most relative gain to that of its partner.
<i>min_diff</i>	Fair —the individual acts in a manner that results in the least disparity.

7.2 Diagnostic Situations

Consider the following scenario: A robot interacts with one of two types of humans, enemies and friendlies, perhaps in a military domain. The relational disposition of

enemies is to consistently use transformations that minimize the robot's outcomes (*min_other*) whereas the transformational tendency of friendlies (*max_other*) is to consistently select transformations that will maximize the robot's outcomes. Can a robot separate enemies from friendlies based solely on their pattern of interaction? We will assume that the robot only has control over its **own** transformation. This section presents foundational material that will result in a solution to this problem.

A diagnostic situation is a situation that reveals a partner's matrix transformation tendencies (Holmes & Rempel, 1989). As such, diagnostic situations can potentially be used to discern a partner's relational disposition. Consider the outcome pattern resulting from interaction between an outcome maximizing robot (*max_own*) and a friendly (*max_other*). Let the transformation pair (*max_own*, *max_other*) represent each member of the dyad's relational disposition. An outcome maximizing robot will consistently select the action that results in the most outcome for itself. A friendly will select the action that will result in the most outcome for its partner—the robot. Hence, after each interaction both individuals will select the action pair which results in the maximum outcome for the robot. Over any number of interactions, the outcome resulting from the transformation pair (*max_own*, *max_other*) will be greater or equal to any other combination of transformation types from Table 7.1 for the robot. Formally the following relationship of outcomes holds $o_1^R > o_2^R$ where the transformation pair (*max_own*, *max_other*) is denoted by the subscript 1 and the transformation pair (*max_own*, *any*) is denoted by the subscript 2.

Alternatively, consider the pattern of outcomes resulting from interaction with an enemy. The transformation pair (*max_own*, *min_other*) will represent each member of the

dyad's relational disposition. In this case, the robot attempts to maximize its own outcome while its partner is attempting to minimize the robot's outcome. The actual outcome received by the robot will depend on a characteristic of the situation called symmetry. Recall from section 6.1, that symmetry describes the degree to which the partners are equally dependent on one another. Importantly, unlike the interaction with the friendly, the robot's outcome will not always be maximal when interacting with an enemy. We can use this fact to discern a pattern of interactions with a friendly partner from a pattern of interaction with an enemy. Figure 7.2 presents an example.

Example: Diagnostic situation

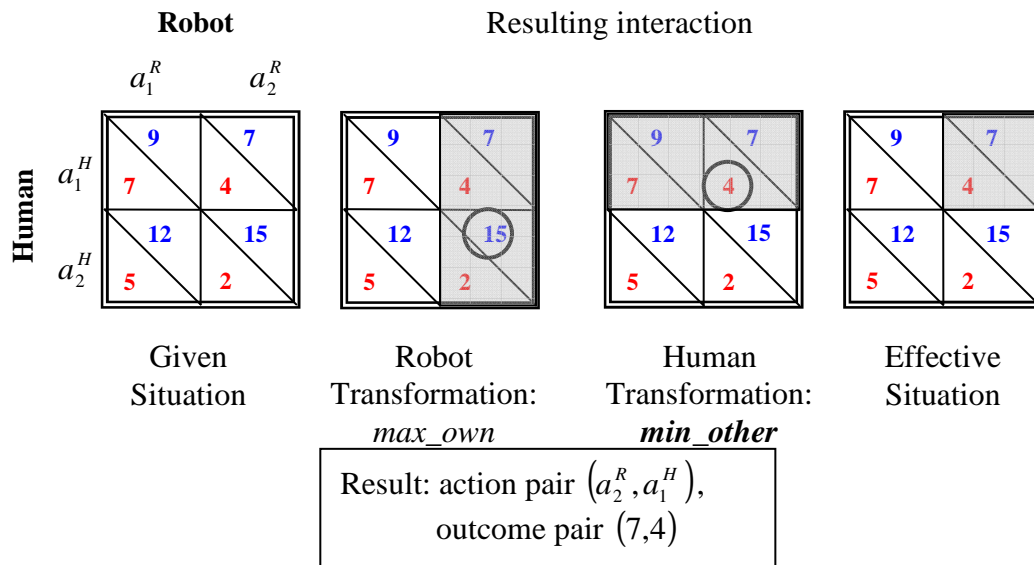
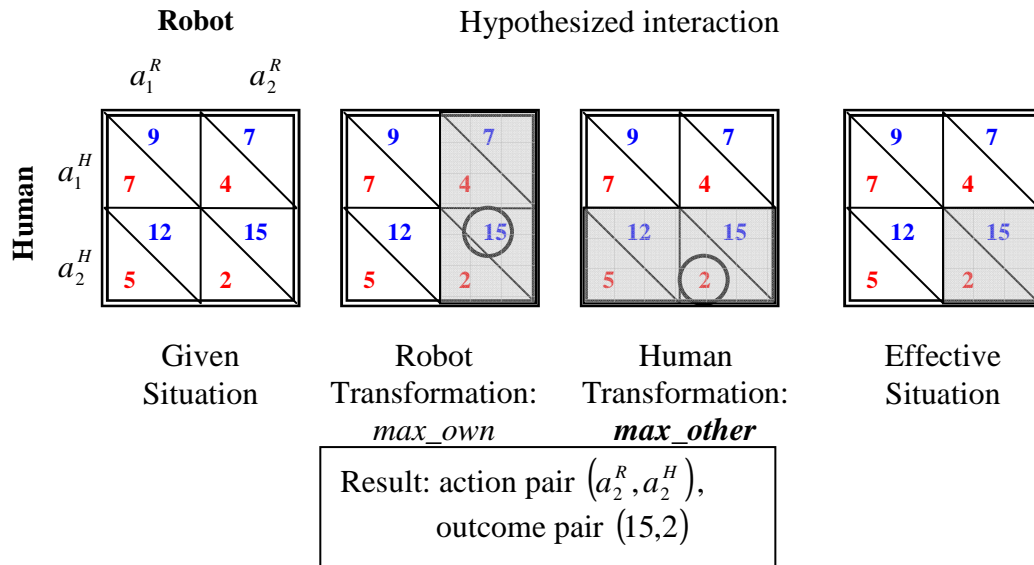


Figure 7.2 An example of a diagnostic situation. The robot and the human are presented with a given situation. The robot selects an action according to a *max_own* transformation and predicts the outcomes resulting for both itself and the human partner if the human selects according to a *max_other* relational disposition. In the resulting interaction depicted below, the human actually selects according to a *max_own* relational disposition. The situation is diagnostic because different outcomes for the robot result from different relational dispositions.

Consider another scenario. Rather than interacting with enemies or friendlies, in this scenario the robot interacts with competitors or cooperators. Competitors attempt to maximize the difference (*max_diff*) in outcome between themselves and the robot whereas cooperators attempt to minimize the difference in outcome (*min_diff*). If we

compare the difference in outcome ($|o^R - o^H|$) resulting from interaction with competitors versus cooperator, here again, the results are clear. In this case, **the difference in outcome** will always be greater or equal when the transformation pair is (max_diff, max_diff) versus (max_diff, min_diff) . Formally the following relationship of outcomes holds $|o_1^R - o_1^H| > |o_2^R - o_2^H|$ where the transformation pair (max_diff, max_diff) is denoted by the subscript 1 and the transformation pair (max_diff, min_diff) is denoted by the subscript 2. In other words, for a given situation, the difference in outcome will never be less when the partner's type is *max_diff* when compared to a partner type of *min_diff*.

We have thus formulated two rules for discerning a partner's type. In the enemies versus friendlies scenario the robot's relational disposition was *max_own* and in the competitors versus cooperators scenario the robot's relational disposition was the *max_diff* transformation. A systematic investigation of each robot relational disposition in Table 7.1 indicates that each robot disposition type can discern between two different partner types given a method of comparison. Table 7.2 lists each robot type with the partner transformation types that it can distinguish as well as the method of comparison. For example, the robot can use the *max_own* transformation type in conjunction with the $o_1^R > o_2^R$ method of comparison to discern a partner of type *max_other* from one that is of type *min_other*. Similarly, the robot can use the *max_diff* transformation type in conjunction with the $|o_1^R - o_1^H| > |o_2^R - o_2^H|$ method of comparison to discern a partner of type *max_diff* from one that is of type *min_diff*. To emphasize the discussion above,

combinations of robot type and methods of comparison will discern particular partner transformation types.

Table 7.2 The table below lists the diagnostic characteristics for different combinations of robot transformation type, first transformation type, second transformation type, and comparator. Each of these combinations does not result in an inverted characterization. The combinations of robot type, hypothesized transformation type and method of comparison only result in diagnostic and non-diagnostic situations and, hence, can be used to determine the partner's relational disposition.

Diagnostic Situation Characterization						
Robot type	Hypoth. Transform. Type	Real Transform. type	Method of Comparison	Diagnostic	Non-diagnostic	Inverted
<i>max_own</i>	<i>max_other</i>	<i>min_other</i>	$o_1^R > o_2^R$	Yes	Yes	No
<i>min_own</i>	<i>min_other</i>	<i>max_other</i>	$o_1^H < o_2^H$	Yes	Yes	No
<i>max_other</i>	<i>max_own</i>	<i>min_own</i>	$o_1^H > o_2^H$	Yes	Yes	No
<i>min_other</i>	<i>min_own</i>	<i>max_own</i>	$o_1^R < o_2^R$	Yes	Yes	No
<i>max_diff</i>	<i>max_diff</i>	<i>min_diff</i>	$ o_1^R - o_1^H > o_2^R - o_2^H $	Yes	Yes	No
<i>min_diff</i>	<i>min_diff</i>	<i>max_diff</i>	$ o_1^R - o_1^H < o_2^R - o_2^H $	Yes	Yes	No
<i>max_joint</i>	<i>max_joint</i>	<i>min_joint</i>	$o_1^R + o_1^H > o_2^R + o_2^H$	Yes	Yes	No
<i>min_joint</i>	<i>min_joint</i>	<i>max_joint</i>	$o_1^R + o_1^H < o_2^R + o_2^H$	Yes	Yes	No

Before moving on to how to discern the difference between partner relational disposition types, we must consider the nature of diagnostic situations more deeply. As mentioned above, a diagnostic situation is a situation that will reveal the partner's transformation type. With respect to Table 7.2 a diagnostic situation occurs whenever the method comparison holds. Thus, for the friendlies and enemies example, a diagnostic situation is any situation in which the outcome resulting from interaction with a friendly is greater than the outcome resulting from the interaction with an enemy. Clearly this should be the majority of situations. Diagnostic situations do not conclusively tell us that our hypothesized partner type is the partner's real type, but they do lend support to the hypothesis. Hence, we cannot conclude that the partner is friendly (*max_other*) simply because the outcome resulting from one interaction with the partner was greater than the

outcome that would have been expected from interaction with an enemy (*min_other*). Figure 7.2 presents an example of a diagnostic situation.

Not all situations are diagnostic however. Many situations do not reveal the partner's transformation type. These situations are termed *non-diagnostic*, as they do not tell us anything about partner's transformation tendencies. The most obvious example of a non-diagnostic situation is a situation that is populated with all of the same outcome values. Returning to our friendlies and enemies example, a non-diagnostic situation will result in the same outcome for the robot regardless of whether the partner type is a friendly or an enemy. Hence, our method of comparison for these type, $o_1^R > o_2^R$, will not hold. Unfortunately, the occurrence of non-diagnostic situation does not tell us that the partner is an enemy and not a friendly. It simply tells us nothing. Figure 7.3 presents an example of a non-diagnostic situation.

Example: non-Diagnostic situation

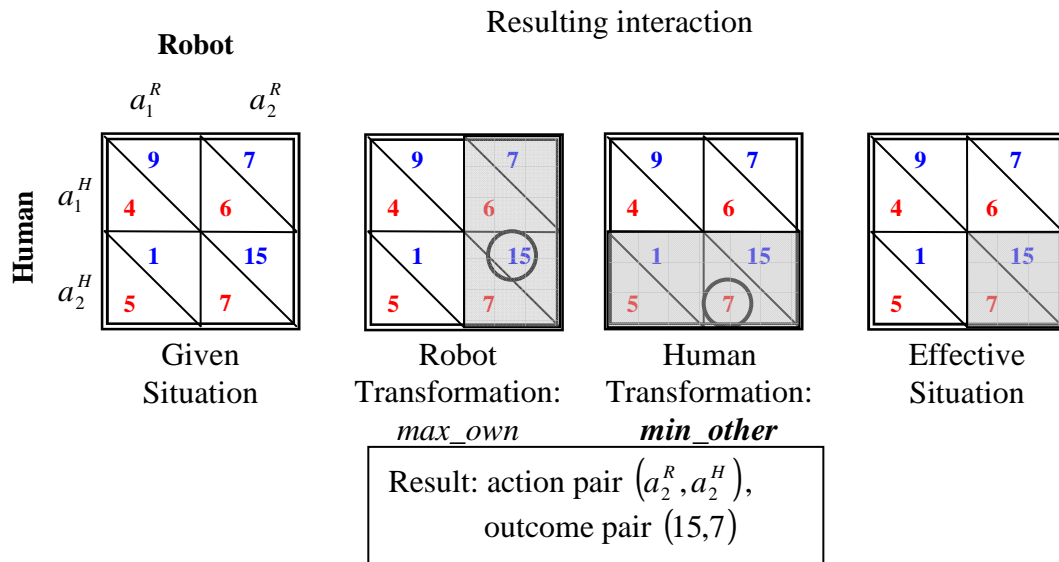
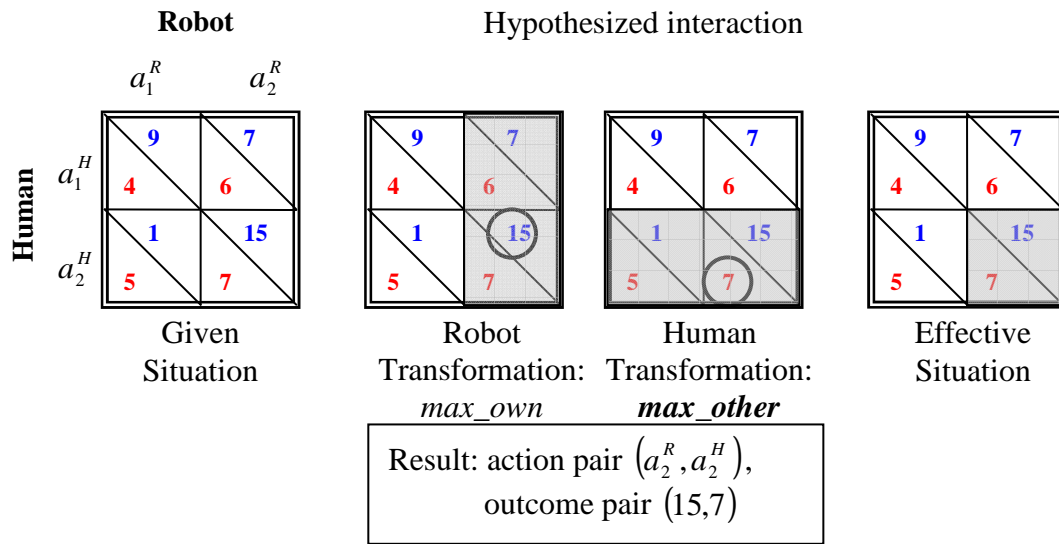


Figure 7.3 An example of a non-diagnostic situation is presented above. The situation is non-diagnostic because the outcome pair is the same regardless of the human's transformation type. The top row presents the interaction hypothesized by the robot and the middle row presents the resulting interaction. The key point here is that this given situation does not distinguish between the human's differing relational dispositions. Even if the robot were to interact with a human in many different non-diagnostic situations, the robot would not be able to determine the human's relational disposition.

If, on the other hand, the pattern of outcomes that results is inverted with respect to our method of comparison, the outcome resulting from interaction with an enemy was greater than that of a friendly, then we can reject our method of comparison. We call this an *inverted* situation. The situation is inverted from our expectations with respect to the

method of comparison. In contrast to a non-diagnostic situation an inversion tells us that our method of comparison and, hence, our hypothesized partner type must be wrong. Considering our example, an inversion tells us that the partner type is not a friendly. Figure 7.4 presents an example of an inverted situation.

Example: Inverted situation

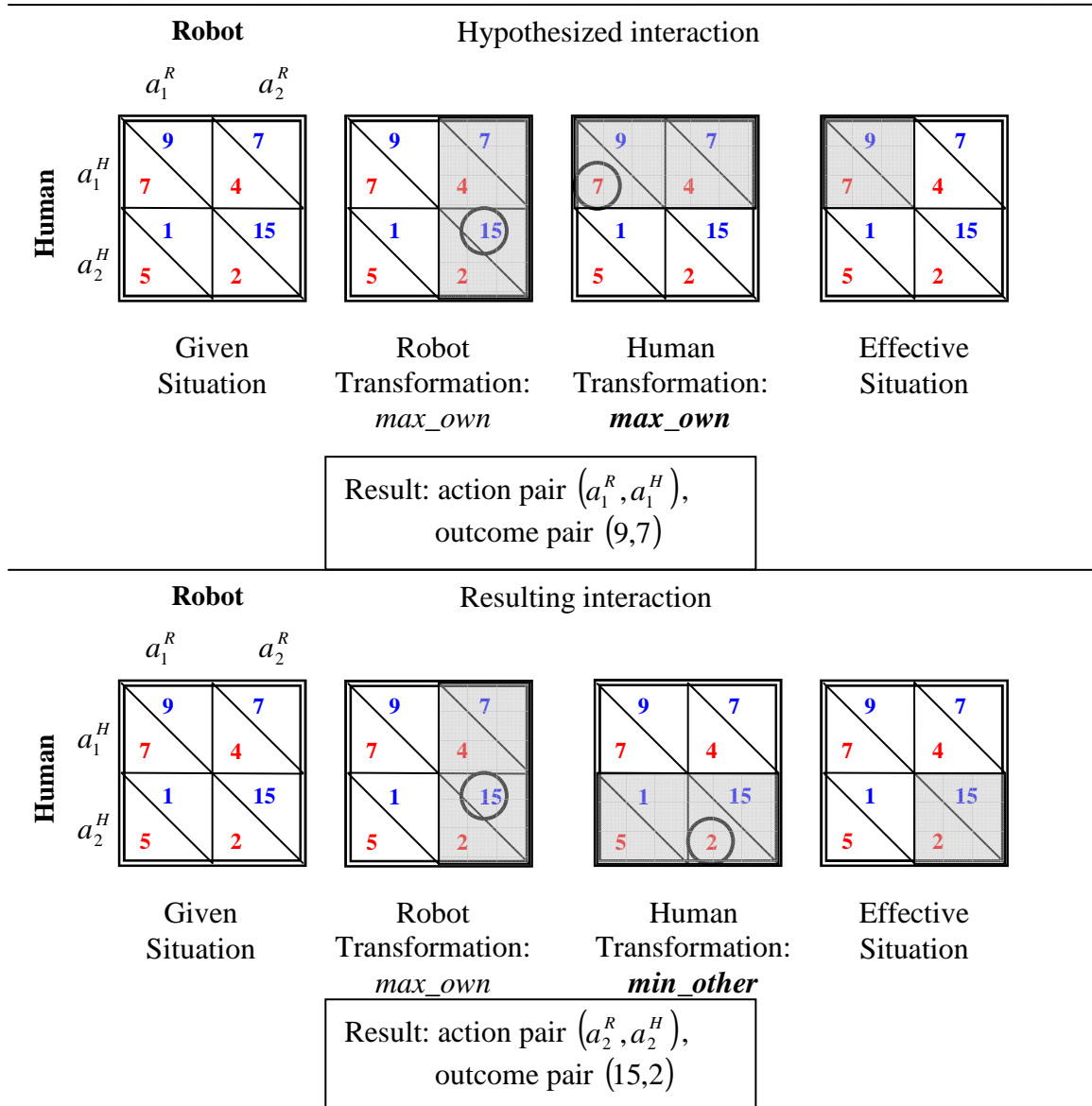


Figure 7.4 An example of an inverted situation. The situation is inverted because the robot's outcome in the resulting interaction (*min_other*) is greater than the robot's outcome in the hypothesized interaction (*max_own*).

Returning to Table 7.2 each of the combinations of robot type, method of comparison and partner types results in both diagnostic and non-diagnostic situations. These combinations do not result in inverted situations. Formally we define the set $Y = \{D, I, N\}$ as representing the three classifications of situation, diagnostic, inverted, and non-diagnostic, respectively.

We now develop an algorithm for determining a situation's classification type given a situation O , a robot transformation type θ^i , and the predicted $(o_k^i, o_k^{-i})^*$ and actual outcomes (o_k^i, o_k^{-i}) of an interaction. It should be noted that much of the preceding discussion described transformation pairs such as (max_own, max_other) and (max_own, min_other) whereas the algorithm uses predicted $(o_k^i, o_k^{-i})^*$ and actual outcomes (o_k^i, o_k^{-i}) . The conversion from transformation pairs to outcomes follows directly by using the function, $O_E^{-i} = f(O_G^{-i}, \theta^{-i})$. Hence, in the examples presented above the following series of steps are used to convert from pairs to transformations to pairs of outcomes,

$$1) f(O_G^{-i}, max_other) = O_E^{-i}$$

$$2) max_own(O_E^{-i}) = a_k^{-i}$$

$$3) f(O_G^{-i}, min_other) = O_E^{-i}$$

$$4) max_own(O_E^{-i}) = a_k^{-i}.$$

5) Finally $O(a_k^i, a_k^{-i}) = (o_k^i, o_k^{-i})$ is used to create the outcome pairs.

The preceding steps represent the partner's transformation process (Figure 2.5 and described in detail in section 4.3). The algorithm (Box 7.1) follows directly from Table

7.2, essentially classifying the situation based on each of the algorithms parameters.

Overall, the parameters to our algorithm are $(O, \theta^i, (o_k^i, o_k^{-i})^*, (o_k^i, o_k^{-i}))$. This algorithm

will be the basis for an algorithm that characterizes the partner's relational disposition.

Figure 7.5 provides an example.

Determining a Situation's Diagnostic Characteristics

Input: Hypothesized interaction result $(o^i, o^{-i})^*$, real interaction results (o^i, o^{-i}) , robot transformation θ^i

Output: Situation's diagnostic characteristics $Y = \{D, I, N\}$

```

1.  if (
    ( $\theta^i = \text{max\_own}$  and  $^*o^i = o^i$ ) or
    ( $\theta^i = \text{max\_other}$  and  $^*o^{-i} = o^{-i}$ ) or
    ( $\theta^i = \text{max\_joint}$  and  $^*o^i + ^*o^{-i} = o^i + o^{-i}$ ) or
    ( $\theta^i = \text{max\_diff}$  and  $|^*o^i - ^*o^{-i}| = |o^i - o^{-i}|$ ))
    //Combinations of robot type and
    //method of comparison resulting in
    //non-diagnostic situation.

2.  return N
3.  else if(
    ( $\theta^i = \text{max\_own}$  and  $^*o^i < o^i$ ) or
    ( $\theta^i = \text{max\_other}$  and  $^*o^{-i} < o^{-i}$ ) or
    ( $\theta^i = \text{max\_joint}$  and  $^*o^i + ^*o^{-i} < o^i + o^{-i}$ ) or
    ( $\theta^i = \text{max\_diff}$  and  $|^*o^i - ^*o^{-i}| < |o^i - o^{-i}|$ ))
    //Combinations of robot type and
    //method of comparison resulting in
    //inverted situation.

4.  return I
5.  else
6.  return D
    //return inverted situation type
    //return diagnostic situation type for all
    //other situations

7.  endif

```

Box 7.1 The algorithm above characterizes situations in terms diagnostic characteristics. The robot type is used to determine the comparator that will be used. Next the outcomes are used in conjunction with the information from Table 7.2 to determine the characterization.

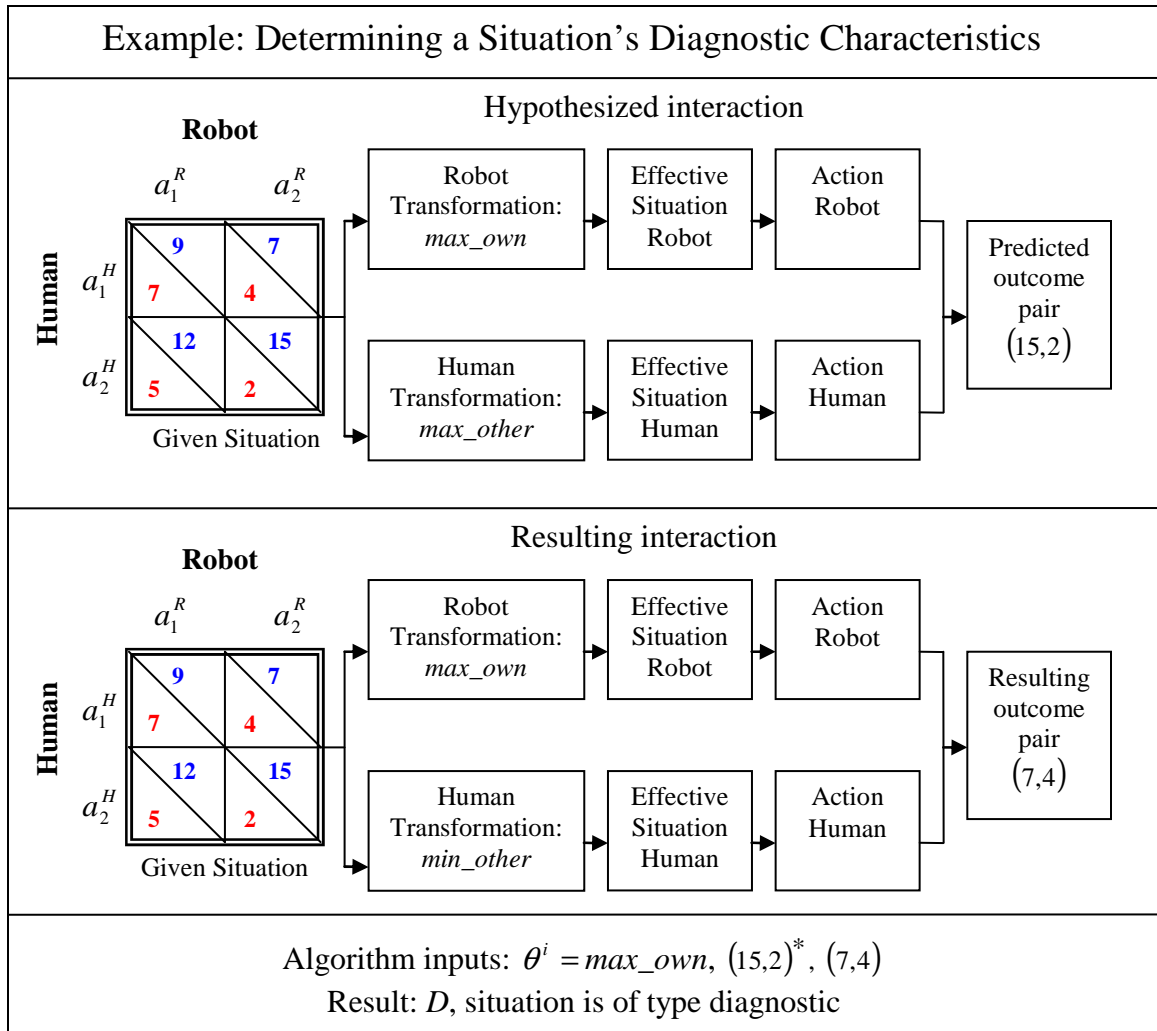


Figure 7.5 The example above uses the given situation from Figure 7.2 and demonstrates use of the algorithm from Box 7.1. The given situation is transformed by the robot and the human to produce an effective situation and finally an action. The action pair results in an outcomes for both the robot and its partner. In the hypothesized interaction (top row) the outcome pair is predicted. In the resulting interaction, the outcome pair is the result of an interaction between the robot and the human. These pairs of outcomes as well as the robot's transformation type are used as input to the algorithm which characterizes the situation as diagnostic.

7.2.1 Diagnostic Situations as a function of Matrix Size

We can use the algorithm presented in previous section to explore the proportion of non-diagnostic to diagnostic situations. A robot that is faced with the challenge of discerning its partner's relational disposition does not generally have complete control over the situation it faces. Rather, it must use whatever situations present themselves to determine

the partner's type. Hence, the existence or lack of diagnostic situations is an important question faced by a robot attempting to discern its partner's relational disposition.

Table 7.3 Summary of the diagnostic situations as a function of matrix size experiment.

Experiment Summary	
Diagnostic Situations as a Function of Matrix Size	
Purpose	Investigate the preponderance of diagnostic situations among matrices of different size.
Experiment Type	Numerical simulation
Hypothesis	As matrix size increases, the ratio of diagnostic to non-diagnostic situations will increase.
Procedure	Follow the procedure presented in Table 7.4.
Independent variable	Matrix size: 2x2, 3x3, 4x4, 5x5, 6x6, 7x7, 8x8, 9x9, 10x10.
Dependent variable	Percentage of diagnostic situations
Method of Analysis	Graph analysis
Conclusion	Hypothesis is supported. As matrix size increases, the number of diagnostic situations increases becoming asymptotic at about size 7x7.

We conducted a numerical simulation to determine how the ratio of diagnostic to non-diagnostic situations changed with respect to the size of the outcome matrix. We reasoned that one important cause of non-diagnostic situations is constriction of individual's action space. Action space constriction occurs when either or both individuals has few actions to choose from in the interaction. This constriction results in a smaller matrix size in terms of the number of columns and/or rows. The smallest matrix that still offers a decision choice for both individuals is a 2x2 matrix. This matrix results in only four pairs of potential outcomes. Hence each combination of transformation pairs is mapped to an outcome pair space of size four. We further reasoned that increasing the action space would increase the relative number of diagnostic situations compared to non-diagnostic situations. Table 7.3 summarizes the experiment.

We used a numerical simulation in this experiment. We tested the hypothesis by presenting a simulated robot and simulated partner matrices of different sizes and

recording the number of diagnostic situations that resulted. To do this, the following procedure was employed:

Table 7.4 Experimental procedure for the diagnostic situations as a function of matrix size experiment.

Experimental Procedure	
1)	Create 1000 matrices of the following different sizes: 2x2, 3x3, 4x4, 5x5, 6x6, 7x7, 8x8, 9x9, 10x10. Each matrix (O) was populated with random values arbitrarily ranging from $[-20,20]$.
2)	For the <i>max_own</i> robot type, the transformation process $O_E^i = f(O, max_own)$; $max_own(O_E^i) = a_k^i$ was used to determine the robot's predicted action. The matching hypothesized transformation for the partner (Table 7.2 row 1 column 2) was used in conjunction with the functions $O_E^{-i} = f(O, \theta^{-i})$; $max_own(O_E^{-i}) = a_k^{-i}$ to determine the partner's action. The predicted outcomes were calculated from $O(a_k^i, a_k^{-i}) = (o^i, o^{-i})^*$.
3)	The same procedure as in the previous step was repeated using the partner's real transformation type (Table 7.2 row 1 column 3). The real outcomes were calculated from $O(a_k^i, a_k^{-i}) = (o^i, o^{-i})$.
4)	Next, the algorithm from Box 7.1 was used to characterize the matrix as either type $\{D, N\}$ diagnostic or non-diagnostic.
5)	The characterization of the situation was recorded.
6)	We repeated the procedure for the <i>max_own</i> , <i>max_other</i> , <i>max_diff</i> , and <i>max_joint</i> rows from Table 7.2

The independent variable for this numerical simulation experiment is matrix size. Hence, we manipulated the action space of both individuals to produce random matrices of a desired size. The dependent variable is the number of diagnostic situations that resulted.

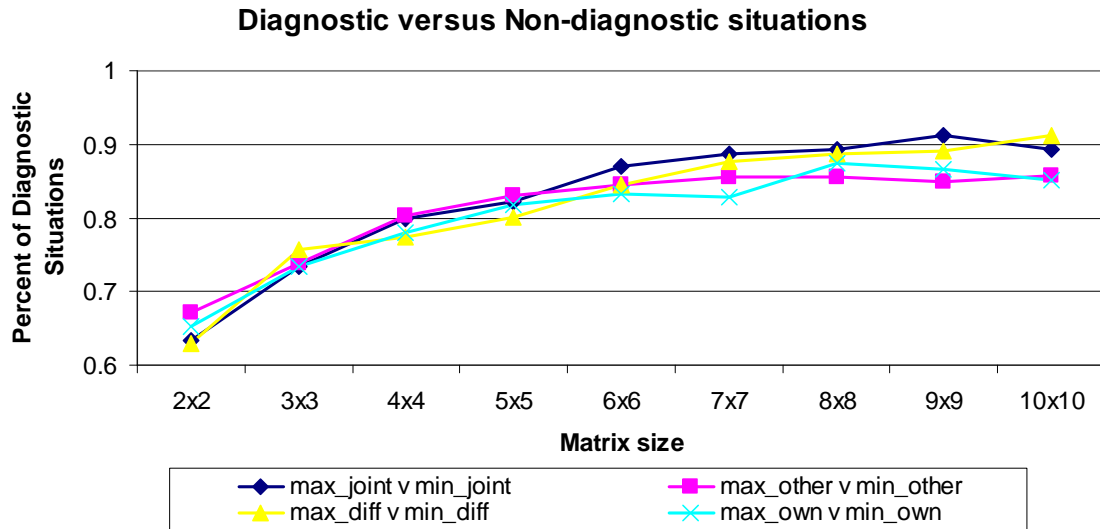


Figure 7.6 The graph above depicts the percentage of diagnostic situations as a function of matrix size. We hypothesized that matrices with fewer actions would result in a smaller percentage of diagnostic situations than matrices with more actions. The trend is true regardless of the type of comparison made.

Figure 7.6 depicts the result. As the matrix size increases from 2x2 to 6x6, the percent of diagnostic situations grows from 65 percent to 86 percent before leveling off at around size 7x7 for each of the robot transformation types. The number of non-diagnostic situations is equal to $1 - n$ where n is the number of diagnostic situations for all data points in the figure. The graph provides support that a constrained action space can limit the robot's ability to determine its partner's relational disposition. For a 2x2 matrix approximately 65 percent of situations are diagnostic, providing information about the partner's relational disposition. For matrices of size 7x7 and greater, this percentage grows to approximately 90 percent.

These results have implications for robot applications in which the robot must determine its partner's relational disposition in few interactions. Military applications involving interaction among both enemy and friendly individuals is one such area. In this case, the robot is better served to have a large action space that will afford a larger proportion of diagnostic situations in general.

7.3 Characterizing Relationships

Diagnostic situations can be used to characterize the relational disposition of an unknown partner. In this section we present an algorithm that utilizes diagnostic situations to determine relational disposition of an unknown interactive partner.

The previous section described diagnostic situations, non-diagnostic situations, and inverted situations. We described a set (rows of Table 7.2) of combinations of robot and partner types along with methods of comparison that resulted in only diagnostic or non-diagnostic types. We noted that the combinations of robot type and partner type along with the method of comparison do not result in inversions.

In this section, we develop an algorithm based on the rows of Table 7.2 that will allow us to determine a partner's transformation type θ^{-i} . In the most general sense, the algorithm operates by hypothesizing a partner type, interacting with the partner over successive situations, and observing the results of the interactions. The algorithm from Box 7.1 is used to classify the situation as diagnostic, non-diagnostic, or inverted. If an inverted situation occurs, then the hypothesized type is rejected. If a non-diagnostic situation occurs then additional interactions with the partner will be necessary. Finally, if a diagnostic situation results, the situation is considered evidence that the hypothesized partner type is the true type. Once the robot's evidence reaches a predefined threshold, the robot concludes that, indeed the hypothesized type is the correct type. Box 7.2 presents the method in the form of pseudocode.

Characterizing a Partner's Relational Disposition

Input: Series of interactions O_1, \dots, O_k

Output: $\theta^{-i} \in T$

1. **For** each $\theta^{-i} \in T$ //For all partner transformation types
2. **For** each O_1, \dots, O_j //For a series of interactions
3. Use Table 7.2 to determine $\theta^i \in T$. //Use the table to determine the robot's type
4. Predict the outcomes that would result from interaction with this type of partner.
5. Interact with the partner to determine the actual outcomes.
6. Use algorithm from Box 7.1 to characterize the situation.
7. Determine if the characterization of the situation allows one to rule out the partner's type or conclude that the hypothesized type is the true type.
8. **Endfor**
9. **Endfor**

Box 7.2 The algorithm above characterizes the partner's relational disposition. It takes as input a series of interactions and outputs the partner's transformation type. The algorithm operates by iterating through each type of relational disposition and several interactions, predicting the outcomes that would result from interaction with the partner type in line 1. After interacting the algorithm in Box 7.1 is used to characterize the situation. The characterization is used to either rule the type out or, possibly, conclude that the hypothesized is the true type.

The algorithm's first step simply iterates through each hypothesized partner transformation type. Next, for each hypothesized partner type the robot interacts in several situations. In step 3, the robot uses Table 7.2 to determine which transformation, $\theta^i \in T$, it should use to test the hypothesized type. Next, the robot uses the partner's transformation process, $O_E^{-i} = f(O, \theta^{-i})$, $max_own(O_E^{-i}) = a_k^{-i}$ to predict the partner's action and its own transformation process to predict its own outcome, $O_E^i = f(O, \theta^i)$, $max_own(O_E^i) = a_k^i$. These actions are used to predict the outcome that will result from the interaction, $O(a_k^i, a_k^{-i}) = (o_k^i, o_k^{-i})^*$. Next, in step 5, the robot interacts with the partner and records the outcomes that result (o_k^i, o_k^{-i}) . Next, the parameters $(O, \theta^i, (o_k^i, o_k^{-i})^*, (o_k^i, o_k^{-i}))$ are used as input to the algorithm for determining a situation's

diagnostic characteristics (Box 7.1). In step 7, the results are used to determine if the characterization of the situation allows one to rule out the hypothesized type if the situation is inverted, or alternatively, conclude that the hypothesized type is the true type if a predefined threshold of diagnostic situations has been met.

The final step of the algorithm assumes the existence of a threshold. The threshold is used to determine if the number of situations characterized as diagnostic or inverted is enough to conclude that the hypothesized type is the partner's true type or to rule out the hypothesis. If we can assume that the partner's relational disposition is constant, then a single characterization of a situation as inverted is enough to rule the relational disposition out. If, on the other hand, the partner's relational disposition is not constant, but rather principally governed by a single transformation type with occasional alternative types, then we can define a ratio of situations characterized as inverted and use this ratio to rule out a hypothesized relational disposition. If the situation is characterized as diagnostic we still cannot conclude that the hypothesized relational disposition is the partner's true relational disposition. Rather, we must define a threshold, either a particular number of diagnostic situations or a ratio of diagnostic situations, in order to conclude that the hypothesized relational disposition is the partner's true relational disposition.

7.3.1 Accuracy of Relational Disposition Algorithm

Much of the previous discussion has assumed that the partner's relational disposition is fixed. In other words, the partner uses a static or fixed transformation during all interactions. This, however, is not realistic. Humans will often alter or dynamically change their relational disposition (Sears, Peplau, & Taylor, 1991). While it is not

uncommon that a human's relational disposition will remain largely static, occasional changes are also normal.

Preliminary experiments involving the algorithm presented in Box 7.2 showed that if the partner's relational disposition is static then the algorithm could determine the partner's type with perfect accuracy. In this case, a single inversion was sufficient to reject a partner type hypothesis, and thresholding the number of diagnostic situations encountered at an arbitrary value of 100 resulted in correct type determination in each of a 1000 attempts.

Table 7.5 Summary of the relational disposition algorithm experiment.

Experiment Summary	
Accuracy of Relational Dispositions Algorithm	
Purpose	Explore the ability of the relational dispositions algorithm to determine the partner's relational disposition.
Experiment Type	Numerical simulation
Hypothesis	As the percent variability of the partner's relational disposition increases, the percent correct determination of the partner's relational disposition will decrease.
Procedure	Follow the procedure presented in Table 7.6.
Independent variable	Percent variability of the partner's relational disposition.
Dependent variable	Percentage correct determination of partner's type.
Method of Analysis	T-test analysis for significance.
Conclusion	Hypothesis is supported. As the percent variability of the partner's disposition type increases the ability of the algorithm to determine the partner's relational disposition decreases.

Still, because a human partner's relational disposition is not expected to remain constant, a more realistic test of the algorithm would allow for occasional variability in the partner's transformation type. We hypothesized that as the amount of partner variability increased the accuracy of the algorithm to determine the partner's type would decrease. Our independent variable was the variability of the partner's relational disposition, which ranged from 0 percent to 20 percent variability. Hence, a single relational disposition was randomly chosen for the partner. That disposition was then

randomly replaced with another relational disposition at a rate determined by the independent variable. Our dependent variable was the accuracy of our algorithm for relational disposition characterization. In other words, the percentage that the algorithm correctly determined the partner's type. Table 7.5 provides a summary of the experiment.

We tested this hypothesis as a numerical simulation. Again, our numerical simulation of interaction focused on the quantitative results of the algorithms and processes under examination and attempts to simulate aspects of the robot, the human, or the environment. The advantages and disadvantages of this approach have already been discussed in section 7.2.1. This numerical simulation experiment again involved a single simulated robot and simulated human. Both selected nominal actions from outcome matrices and received the outcomes that resulted, but the actions were not performed by either individual.

To test this hypothesis the following procedure was followed:

Table 7.6 Experimental procedure for the relational disposition algorithm experiment.

Experimental Procedure	
1)	One thousand situations populated with random values (arbitrary range of $[-20,20]$ was used) were created.
2)	The robot's partner was assigned a random "target" relational disposition.
3)	The robot is presented with one outcome matrix (O) from the thousand.
4)	The robot performs steps 1-4 of our algorithm for characterizing a partner's relational disposition (Box 7.2).
5)	For step 5 of our algorithm for characterizing a partner's relational disposition, the robot uses its relational disposition, θ^i , to select an action (a_k^i) according to the transformation process $O_E^i = f(O, \theta^i); \max_{own}(O_E^i) = a_k^i$. The partner also uses its relational disposition, θ^{-i} , and transformation process, $O_E^{-i} = f(O, \theta^{-i}); \max_{own}(O_E^{-i}) = a_k^{-i}$, to select an action.
6)	The resulting outcomes, from (o^i, o^{-i}) are calculated from the action pair and the outcome matrix, $(O(a_k^i, a_k^{-i}))$.
7)	The robot performs steps 6 and 7 of our algorithm for characterizing a partner's relational disposition (Box 7.2).
8)	Steps 3-7 are repeated until our algorithm has determined the partner's relational disposition.
9)	The partner relational disposition returned by the algorithm is compared to the partner's true relational disposition and the result is recorded.

With each new situation faced by the dyad, the partner's relational disposition was selected in accordance with the independent variable. Hence, if the independent variable was set to explore the results of using the algorithm with a partner that varies their disposition in 7 percent of situations, then the partner's disposition had a 7 percent chance of being different from the base type with each interaction.

Step 7 of our algorithm determines if the situation's characterization allows us to rule out the partner type hypothesis or conclude that the hypothesis is true. Because the partner has a non-constant relational disposition, it is not possible to reject a partner type hypothesis because of a single inversion. Rather, we experimented with different ratios

for rejecting or accepting the partner type hypothesis. Letting d be the number of diagnostic characterizations, n the number of non-diagnostic characterizations, v the number of inverted characterizations, and T be the total number of all characterizations, the conditions for rejecting a partner type hypothesis in this experiment was set to $\frac{v}{T} > 0.04$ and $T > 10$. The conditions for accepting a partner type hypothesis was set to

$\frac{n+d}{T} > 0.04$ and $T > 100$. These values were empirically derived.

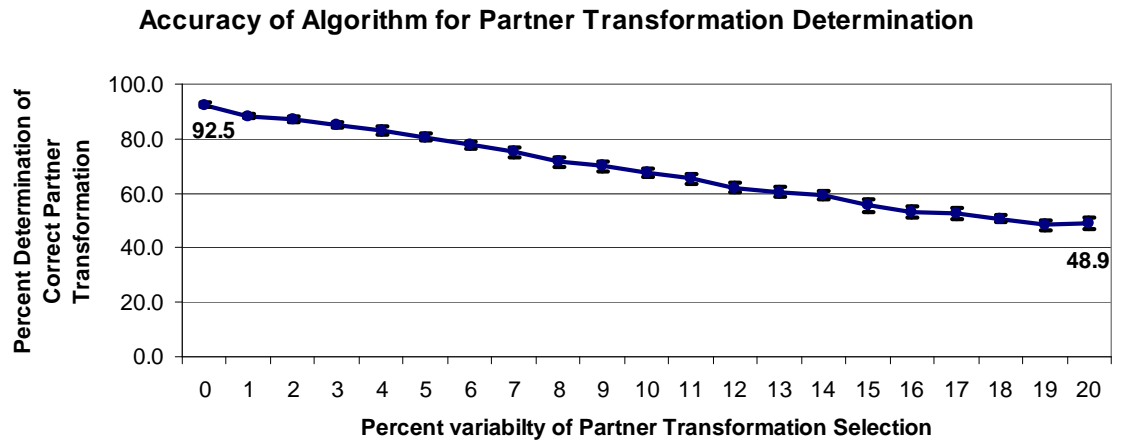


Figure 7.7 The graph above depicts the accuracy of the partner’s relational disposition as a function of partner transformation type variability. We hypothesized that as the partner’s transformation variability increased the algorithm’s accuracy would decrease. The results above support our hypothesis.

Figure 7.7 presents the results of the experiment. As hypothesized the overall accuracy of algorithm decreases as the variability of the partner’s transformation selection increases. The rate of decrease is approximately linear. Yet the slope of decrease is $\frac{48.9 - 92.5}{20} = -2.18$. When the partner’s relational disposition did not vary,

the algorithm, using the ratios for accepting and rejecting a partner hypothesis described above, was successful 92.5 percent of the time. When the partner’s transformation type

varied 20 percent of the time, the algorithm was only 48.9 percent accurate. Each data point in Figure 7.7 also displays the point's confidence interval.

The results of the numerical simulation indicate that the accuracy of our algorithm for determining a partner's relational disposition decreases rapidly with partner variability (slope of decrease is $\frac{48.9 - 92.5}{20} = -2.18$). Domain specific testing will be necessary to determine if the reduction in accuracy will inhibit task or domain specific performance. For example, warfare domains may result in minimal (less than 5%) partner transformation variability. Still, even with this low amount of variability, an accuracy of ≈ 80 may be insufficient if the result of being characterized as an enemy is being fired upon.

Transformation similarity may also make the results appear worse than they actually are. For instance, the *max_joint* transformation has similar outcome and action characteristics as the *min_diff* transformation. Hence, a coarser division of transformations as pro and antisocial could potentially result in much better algorithm performance.

Alternatives to this algorithm, such as standard machine learning techniques, may also improve performance. The use of support vector machines or other machine learning techniques could potentially outperform the presented algorithm. The advantage of the algorithm we present is that its performance is not based on training data and hence affords relational disposition determination without being first trained to do so. It may not be possible to train a robot to determine their partner's relational disposition in every environment they will face, hence the value of the algorithm we propose.

7.4 Conclusions

The preceding chapter has begun the long and challenging task of investigating how a robot should represent and reason about its relationships. Using interdependence theory as an underpinning, we have argued that relationships accrete from a series of interactions and that these interactions result in a pattern of outcomes which can be used to characterize the relationship. We have developed algorithms based on this pattern of outcomes that allow a robot to determine its partner's relational disposition. Our algorithm is based on the robot's ability to characterize a situation as diagnostic, non-diagnostic, or inverted.

The experiments presented in this chapter are much more of an introduction to the study of human-robot relationships than a conclusion. We have examined the hypothesis that a constrained action space is one cause for non-diagnostic situations. Our results indicate that, indeed, matrix size is a factor for determining the proportion of non-diagnostic to diagnostic situations. We have also examined an algorithm that characterizes a partner's relational disposition. Our results here indicate that the accuracy of the algorithm's determination of partner type decreases rapidly with increased partner type variability.

Overall, the research presented in this chapter represents a novel and interesting approach to the exploration of human-robot relationships. We have developed the first algorithms allowing a robot to discern and characterize its relationships and have illuminated aspects of this topic of research which may prove critical for human-robot relationship understanding.

CHAPTER 8

TRUST IN HUMAN-ROBOT INTERACTIONS

Trust. The term itself conjures vague notions of loving relationships and lifelong familial bonds. But is trust really so indefinable? As we detailed in the second chapter, the phenomena of trust has been seriously explored by numerous researchers for decades. Moreover, the notion of trust is not limited to interpersonal interaction. Rather, trust underlies the interactions of employers with their employees, banks with their customers, and of governments with their citizens. In many ways trust is a precursor to a great deal of normal interpersonal interaction.

For interactions involving humans and robots, an understanding of trust is particularly important. Because robots are embodied, their actions can have serious consequences for the humans around them. Several people have already died as a result of their work with robots (Economist, 2006). A great deal of research is currently focused on bringing robots out of labs and into people's homes and workplaces. These robots will interact with humans—such as children and the elderly—unfamiliar with the limitations of a robot. It is therefore critical that human-robot interaction research explore the topic of trust.

In contrast to much of the prior work on trust, the research presented here does not begin with a model for trust. Rather, we begin with a very simple idea: *if it is true that outcome matrices serve as a representation for interaction, then should it not also be true that some outcome matrices include trust while others do not?* In other words, some interpersonal interactions require trust, yet others do not. If an outcome matrix can be

used to represent all interactions then it should also represent those interactions which require trust. Our task then becomes one of determining what the conditions for trust are.

8.1 Recognizing Situations that Require Trust

We begin the task of delineating the conditions for trust with a definition. As first introduced in section 2.3.1 *trust is a belief, held by the trustor, that the trustee will act in a manner that mitigates the trustor's risk in a situation in which the trustee has put its outcomes at risk.*

Rather than recognizing interactions that require trust, we will present conditions for recognizing situations that require trust. Recall that social situations abstractly represent a class of interactions. This section develops conditions for classifying a situation in terms of trust. Classification of a situation in terms of trust is a binary task, i.e. a true/false statement concerning whether or not the selection of an action in a situation would require trust. The section that follows introduces a method for measuring trust.

Consider, for example, the trust fall. The trust fall is a game played in an attempt to build trust between two or more people. One person simply leans backward and falls into the awaiting arms of another person (Figure 8.1). We will use the trust fall as a running example to explain our conditions for trust.

Assume that the trust fall involves two people. The person leaning back acts as the trustor, whereas the person doing the catching represents the trustee. The trustor decides between two potential actions in the interaction: lean back and do not lean back. The trustee also decides between two potential actions: catch the falling person and do not catch the falling person. Hence we can represent the interaction as a 2x2 outcome matrix (Figure 8.1). In this interaction the trustor holds the belief that the trustee will break their

fall before they hit the ground. Moreover, the action of leaning back puts the trustor at risk of possible injury. The actual result of the interaction depends on the actions of the trustee. Their choice of action can result in injury or no injury to the trustor. As described, the situation implies a specific pattern of outcome values.



Figure 8.1 An example of the trust fall. The trust fall is a trust and team-building exercise in which one individual, the trustor, leans back prepared to fall to the ground. Another individual, the trustee, catches the first individual. The exercise builds trust because the trustor puts himself at risk expecting that the trustee will break her fall.

The definition for trust listed above focuses on the actions of two individuals: a trustor and a trustee. These individuals can be arbitrarily listed as the interacting individuals in an outcome matrix (Figure 8.2). Without loss of generality, we limit our discussion of the decision problem to two actions (a_1^i and a_2^i for the trustor, a_1^{-i} and a_2^{-i} for the trustee). We will arbitrarily label a_1^i as the trusting action and a_2^i as the untrusting action for the trustor. Similarly, for the trustee the action a_1^{-i} arbitrarily denotes the

action which maintains trust and the action a_2^i the action which does not maintain trust. The definition for trust implies a specific temporal pattern for trusting interaction. Because the definition requires risk on the part of the trustor, the trustor cannot know with certainty which action the trustee will select. It therefore follows that 1) *the trustee does not act before the trustor*. We can model this condition in outcome matrix notation as $i \Rightarrow -i$.

Situational Trust with Example

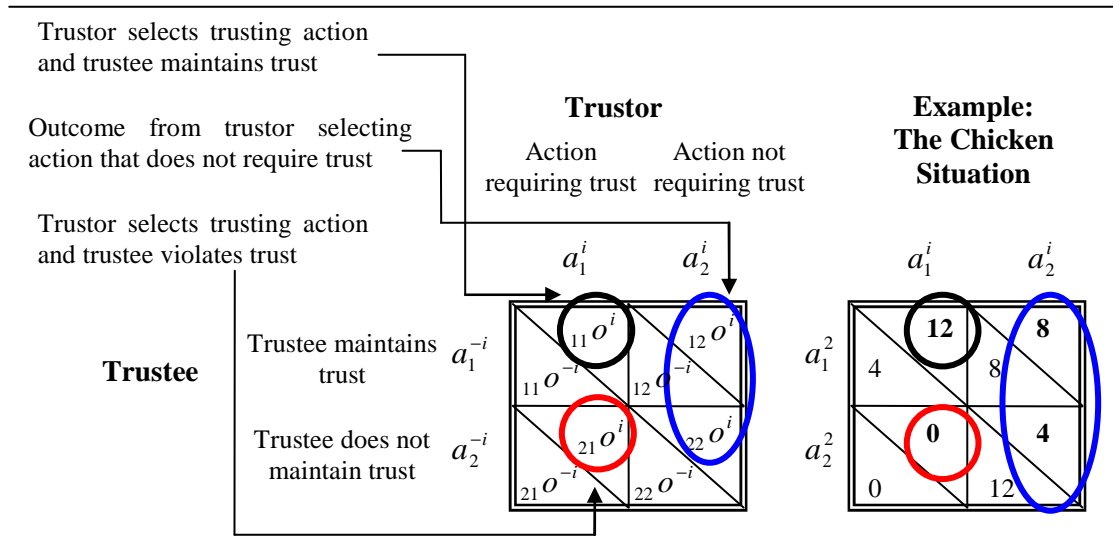


Figure 8.2 The figure visually depicts the reasoning behind the development of the conditions for trust. The matrix on the left visually describes the conditions for trust. The matrix on the right presents an example.

The definition for trust notes that risk is an important consideration for the trustor. Risk refers to a potential loss of outcome. The occurrence of risk implies that the outcome values received by the trustor depend on the action of the trustee. Our second condition notes this dependence relation by stating that 2) *the outcome received by the trustor depends on the actions of the trustee if and only if the trustor selects the trusting action*. Recall that o^i denotes the outcome received by the trustor. If the trustor selects the trusting action then we are comparing the outcomes $_{11}o^i$ and $_{21}o^i$ from Figure 8.2.

The statement indicates that there will be a difference, ${}_{11}o^i - {}_{21}o^i > \varepsilon_1$, where ε_1 is a constant representing the minimal amount for dependence, between these two outcomes.

The trustor may also select the untrusting action, however. The existence of the untrusting action implies that this action does require risk on the part of the trustor. In other words, the outcome $({}_{x2}o^i)$ received by the trustor when selecting the untrusting action does not depend on the actions of the trustee. This leads to a third condition, 3) *the outcome received when selecting the untrusting action does not depend of the actions of the trustee*. In other words, the outcomes for action a_2^i do not depend on the action selected by the trustee. Stated formally, $|{}_{12}o^i - {}_{22}o^i| < \varepsilon_2$, where ε_2 is a constant representing the maximal amount for independence between these two outcomes.

The definition for trust implies a specific pattern of outcome values. It indicates that the trustor is motivated to select the trusting action only if the trustee mitigates the trustor's risk. If the trustee is not expected to mitigate the trustor's risk then it would be better for the trustor to not select the trusting action. We can restate this as a condition for trust, 4) *the value, for the trustor, of fulfilled trust (the trustee acts in manner which mitigates the risk) is greater than the value of not trusting at all, is greater than the value of having one's trust broken*. Again described formally, the outcomes are valued ${}_{11}o^i > {}_{x2}o^i > {}_{21}o^i$.

Finally, the definition demands that, 5) *the trustor must hold a belief that the trustee will select action a_1^{-i} with sufficiently high probability*, formally $p^i(a_1^{-i}) > k$ where k is some sufficiently large constant.

The preceding conditions are necessary for a trusting interaction. Sufficiency occurs when these conditions are met and the trustor selects action a_1^i . The first four conditions describe the situational conditions necessary for trust. By testing a situation for these conditions one can determine whether or not an interactive situation requires trust. Box 8.1 presents an algorithm for determining if a putative situation requires trust.

Testing for Situational Trust

Input: Outcome matrix O

Assumptions: Individual i is trustor, individual $-i$ is trustee, action a_1^i is the trusting action and action a_2^i is not a trusting action.

Output: Boolean stating if O requires trust on the part of individual i .

1. **If** $i \Rightarrow -i$ is not true //the trustee does not act before the trustor
Return false
2. **If** ${}_{11}o^i - {}_{21}o^i > \varepsilon_1$ is not true //the outcome received by the trustee depends on the
Return false //action of the trustor when selecting the trusting action
3. **If** ${}_{12}o^i - {}_{22}o^i < \varepsilon_2$ is not true //the outcome received by the trustee does not depend on
Return false //the action of the trustor when selecting the untrusting action
4. **If** ${}_{11}o^i > {}_{x2}o^i > {}_{21}o^i$ is not true //the value of fulfilled trust is greater than the value of not
Return false //trusting at all, is greater than the value of having one's trust
Else // broken.
Return true

Box 8.1 The algorithm above depicts a method for determining whether a social situation requires trust. The algorithm assumes that the first individual is the trustor, the second individual is the trustee, the action a_1^i is the trusting action, and the action a_2^i is not a trusting action.

The chicken situation (see Figure 8.2) is an example of a social situation that potentially meets the conditions for situational trust. The first condition will be assumed to be true. In this situation, the second condition results in values (from the matrix) $12 - 0 > \varepsilon_1$. Thus, action a_1^i does depend on the actions of the partner for constant $\varepsilon_1 < 12$. The values assigned to the constants $\varepsilon_1, \varepsilon_2, k$ are likely to be domain specific. The constant ε_1 represents a threshold for the amount of risk associated with the trusting

action. The constant ε_2 , on the other hand, represents a threshold for the lack of risk associated with the untrusting action. The third condition results in values, $|8 - 4| < \varepsilon_2$. Here, the action a_2^i does not depend on the actions of the partner for constant $\varepsilon_2 \geq 4$. The final condition results in values $12 > \{8, 4\} > 0$. Hence, for individual one, the selection of action a_1^i involves risk that can be mitigated by the actions of the partner and the selection of action a_2^i does not involve risk that is mitigated by the actions of the partner. Appendix B lists additional situations that meet the conditions for situational trust under the assumption $i \Rightarrow -i$ or $i \Leftrightarrow -i$.

8.1.1 Interdependence space mapping of situational trust

The preceding discussion has introduced a method for testing whether or not a situation demands trust. In this section we use this method to test a hypothesis about the nature of trust itself.

Table 8.1 Summary of the interdependence space mapping of situational trust experiment.

Experiment Summary	
Interdependence space mapping of situational trust	
Purpose	Determine if situations meeting the conditions for situational trust occupy a particular portion of the interdependence space.
Experiment Type	Numerical simulation
Hypothesis	Situations which do meet the conditions for trust do carve out a subspace of the interdependence space.
Procedure	Follow the procedure presented in Table 8.2.
Independent variable	Whether or not a particular situation meets the conditions for trust.
Dependent variable	Location within the interdependence space.
Method of Analysis	Graph analysis.
Conclusion	Hypothesis is supported. Situations meeting the conditions for trust do not create a subspace within the interdependence space.

Given that all social situations occupy some location in the interdependence space, we considered the possibility that situations demanding trust carve out separate portion of

that space apart from those situations not demanding trust. For example, if all situations demanding trust occur in a limited location of the interdependence space, then the interdependence properties of that location could provide important information as to the nature of trust itself. We therefore hypothesized that those situations that meet the conditions for trust would indeed carve out a subspace of the interdependence space. Table 8.1 summarizes the experiment.

Our hypothesis was based, in part, on our experience investigating the interdependence conditions necessary for deception. In research currently under review we have demonstrated that social situations that warrant the use of deception form a subspace of the interdependence space. Hence, it seemed reasonable to expect that the situations that met the conditions for trust would similarly form a subspace within the interdependence space.

We used a numerical simulation to test this hypothesis. The following experimental procedure was followed:

Table 8.2 Experimental procedure for the interdependence space mapping of situational trust experiment.

Experimental Procedure	
1)	Create 1000 matrices populated with random values arbitrarily ranging from [0,24].
2)	Our algorithm Box 6.1 was used to determine the situation's location in interdependence space. The situation's location was recorded.
3)	Each situation was used as an input to the algorithm in Box 8.1 and the result was recorded.

Figure 8.3 depicts the results. The graph on the left hand side depicts those situations which our algorithm for situational trust indicates as demanding trust. The graph on the right hand side depicts all 1000 situations with those demanding trust colored red and those not demanding trust colored blue. In the right hand side, notice that the situations

demanding trust are interspersed throughout with the situations not demanding trust. Figure 8.4 depicts the same graph in two dimensions for easier understanding. Again we see approximately the same scattering regardless of the conditions for trust. We conclude that the conditions for trust do not result in a subspace of the interdependence space.

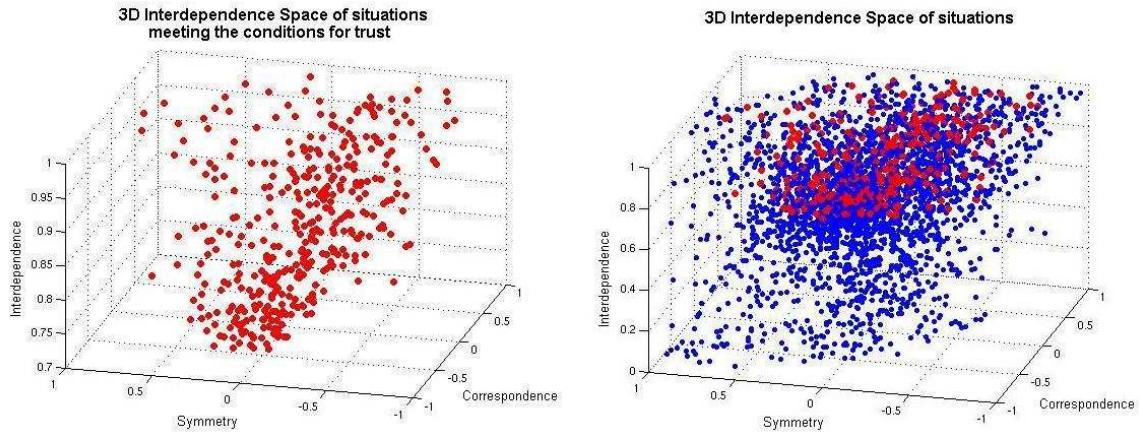


Figure 8.3 The graphs depict the interdependence space mapping of random situations. The left hand side depicts only the situations meeting the conditions for trust (red). The right hand side depicts both those situations meeting the conditions for trust and those not meeting the conditions (blue). We hypothesized that the situations meeting the conditions for trust would form a subspace in the right hand side graph. As can be seen, the situations meeting the conditions for trust are interspersed with situations not meeting the conditions. Hence, our hypothesis is false; the situations meeting the conditions for trust do not form a subspace of the interdependence space.

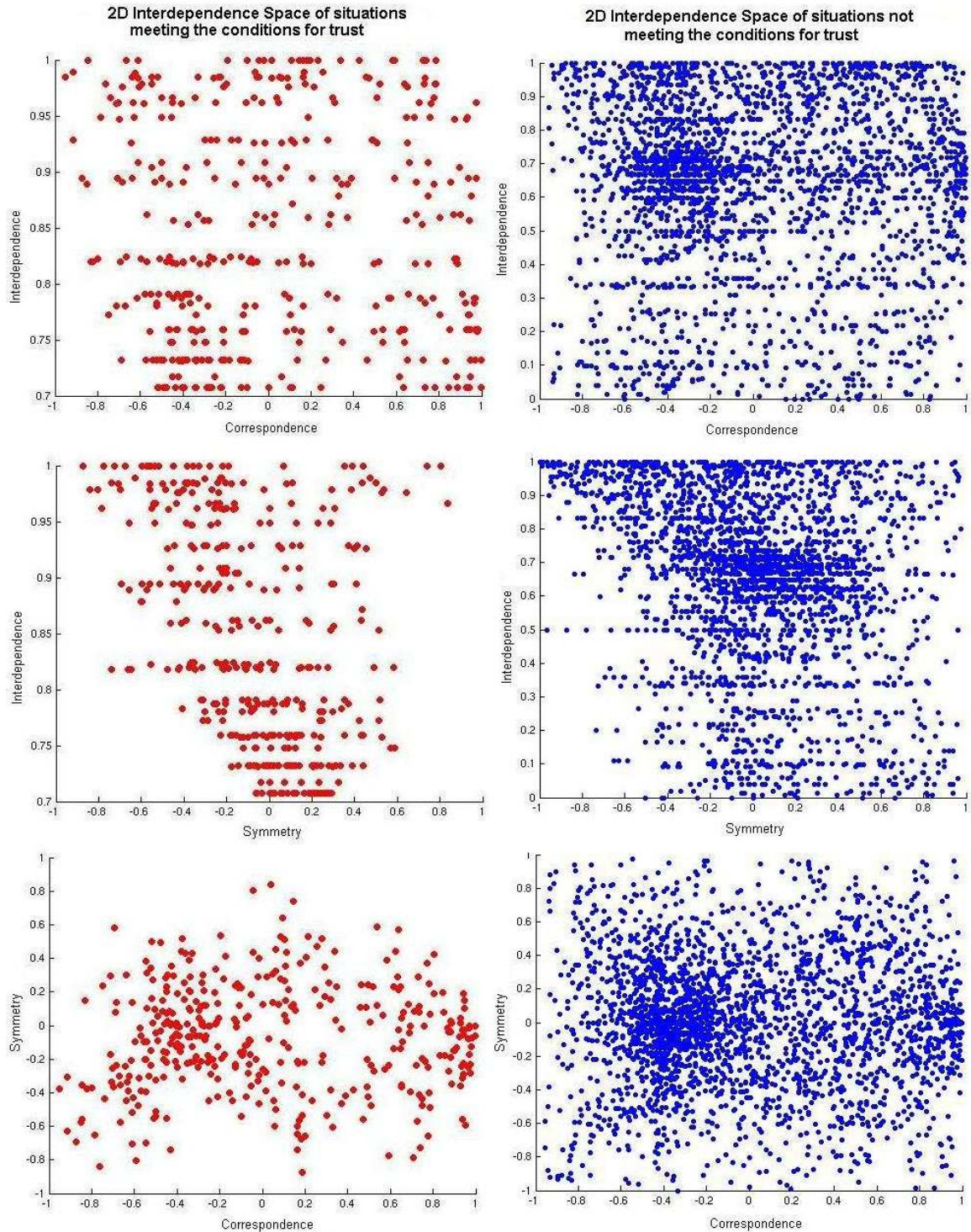


Figure 8.4 The figure depicts 2D graphs of situations meeting the conditions for trust (left hand side) and situations not meeting the conditions for trust (right hand side). Comparison of the graphs to the right with the graphs on the left indicates no difference. Hence, in none of the 2D graphs does the space of situations meeting the conditions for trust form a subspace of the interdependence space separate from those situations that do require trust.

Looking closer at the conditions for trust, the reasons become apparent. The conditions for trust place conditions on the pattern of outcomes for only the trustor but not for the trustee. Yet a situation's mapping in the interdependence space results from the pattern of outcomes for **both** individuals. Hence, while the trustor's pattern of outcomes is not random, the trustee's is random. Overall, the combination of the trustor and trustee's patterns of outcomes does not carve a subspace of the interdependence space. Hence, interdependence space conditions cannot be used to test or to measure the amount of trust.

8.1.2 Canonical situations and the conditions for trust

But do our conditions for trust agree with normative human understanding of trust? In other words, does the Testing for Situational Trust algorithm (Box 8.1) select the same situations as requiring trust as would a human or group of humans? We hypothesize that it does.

Table 8.3 Summary of the canonical situations and the conditions for trust experiment.

Experiment Summary	
Canonical situations and the conditions for trust	
Purpose	Provide evidence that the conditions for trust agree with normative human understanding of trust.
Experiment Type	Numerical simulation
Hypothesis	Our conditions for trust select the same situations as requiring trust as would a human or group of humans.
Procedure	The following procedure was used: 1) Select situations to examine. 2) Input into the algorithm from Box 8.1. 3) Record algorithm's output.
Independent variable	Situations tested.
Dependent variable	Determination of whether or not the situation demands trust.
Conclusion	Experiment provides evidence in support of the hypothesis. Several situations commonly associated with trust are judged as demanding trust by our algorithm. Similarly, situations not expected to require trust are judged not to require trust by our algorithm.

In this section we qualitatively compare examples of those situations which meet the conditions for trust to those which do not. Our goal is to demonstrate that the situations selected by our algorithm as demanding trust intuitively match those situations in which humans use trust. Additionally, we strive to show that situations which are typically not considered to demand trust are also deemed to not require trust by our algorithm. The purpose of this analysis is to provide support for the hypothesis that the Testing for Situational Trust algorithm (Box 8.1) does relate to the conditions underlying normative interpersonal trust. It is challenging, if not impossible, to show conclusively outside of a psychological setting that indeed our algorithm equates to normal human trust processes. Table 8.3 summarizes the experiment.

In order to test this hypothesis, we selected five situations listed in Kelly et al.'s atlas of social situations (Kelley et al., 2003). Table 8.4 lists the five social situations. The situations were selected because they represent different areas of the interdependence space. Each situation was used as input to the algorithm in Box 8.1. The values for constants were arbitrarily set at $\varepsilon_1 = 6$ and $\varepsilon_2 = 6$. The independent variable is the situations selected for testing. The dependent variable then is the determination of whether or not the situation demands trust.

The results are listed in the rightmost column of Table 8.4. This column states whether or not the algorithm indicates that the situation demands trust on the part of the trustor. The trustor is assumed to be the individual depicted on the top of the matrix. The trusting action is assumed to be located in the first column of each matrix.

For example consider the Cooperative Situation, the first row from Table 8.4. The outcome matrix for the situation is used as input to the algorithm. The first line in the

algorithm is assumed to be true. The second line of the algorithm calculates ${}_{11}o^i - {}_{21}o^i > \varepsilon_1$ as $13 - 6 > 6$. Hence the second condition for situational trust is true. The third line of the algorithm calculates $|{}_{12}o^i - {}_{22}o^i| < \varepsilon_2$ as $|6 - 6| < 6$. This third condition for situational trust is also found to be true. Finally, the fourth line of the algorithm computes ${}_{11}o^i > {}_{x2}o^i > {}_{21}o^i$ to be $13 > 6, 6 > 6$ which is false. The final output of the algorithm for this situation is false.

Table 8.4 Several situations arbitrary situations are depicted above. The table includes a description of the situation and the situation's outcome matrix. The first condition the algorithm in Box 8.1 is assumed to hold for all situations. Columns 3-5 present the results for the remaining conditions. The right most column presents the algorithm's final output, stating whether or not the situation demands trust.

Social Situations for Qualitative Comparison						
Situation	Outcome Matrix		Condition 2	Condition 3	Condition 4	Situational Trust?
Cooperative Situation — Each individual receives maximal outcome by cooperating with the other individual.	13	6	True	True	False	False
	12	6				
	6	6	True	True	False	False
	6	0				
Competitive Situation —Each individual gains from the other individual's loss. Maximal outcome is gained through non-cooperation.	6	12	False	False	False	False
	6	0				
	0	6	False	False	False	False
	12	6				
Trust Situation —In this situation, cooperation is in the best interests of each individual. If, however, one individual suspects that the other will not cooperate, non-cooperation is preferred.	12	8	True	True	True	True
	12	0				
	0	4	True	True	True	True
	8	4				
Prisoner's Dilemma Situation —Both individuals are best if they act non-cooperatively and their partner acts cooperatively. Cooperation and non-cooperation, results in middling outcomes for both.	8	12	True	False	False	False
	8	0				
	0	4	True	False	False	False
	12	4				
Chicken Situation —Each individual chooses between safe actions with middling outcomes and risky actions with extreme outcomes.	12	8	True	True	True	True
	4	8				
	0	4	True	True	True	True
	0	12				

The following additional situations were analyzed:

1. The Cooperative situation describes a social situation in which both individuals interact cooperatively in order to receive maximal outcomes. Given the algorithm's parameters, the trustor faces a situation in which the trusting action is dependent on the trustee. The untrusting action, in contrast, is not dependent on the trustee. Nevertheless, the trustor stands to lose nothing if the trustee does not maintain trust (6 versus 6). Hence, selection of the trusting action does not involve risk as the trustor stands to minimally gain as much by selecting this action as by selecting the untrusting action. We therefore conclude that the situation does not meet the conditions for trust.

2. The Competitive situation also does not demand trust, but for different reasons. In this situation the trusting and untrusting actions afford equal risk. Thus the trustor does not face a decision problem in which it can select an action that will mitigate its risk. Rather, the trustor's decision problem is simply of a matter of selecting the action with the largest guaranteed outcome. Trust is unnecessary because the trustor's decision problem can be solved without any consideration of the trustee's beliefs and actions.

3. The Trust Situation describes a situation in which mutual cooperation is in the best interests of both individuals. As the name would portend, this situation demands trust. The trustor's outcomes are dependent on the action of the trustee if it selects the trusting action. Further, nominal outcomes are risked when selecting untrusting action. Finally, the trustor stands to gain the most if it selects the trusting action and the trustee maintains the trust. The trustor's second best option is not to

trust the trustee. Finally, the trustor's worst option is to select the trusting action and to have the trustee violate that trust.

4. The Prisoner's Dilemma is perhaps the most extensively studied of all social situations (Axelrod, 1984). In this situation, both individual's depend upon one another and are also in conflict. In this situation, selection of the trusting action by the trustor does place outcomes at risk dependent on the action of the trustee. Given the parameters selected, however, the untrusting action is also critically dependent on the action of the trustee. Hence, the decision problem faced by the trustor is more complicated than simply dissecting the problem into trusting and untrusting actions. Importantly, both actions require some degree of risk on the part of the trustor. Our conditions for situational trust demand that the decision problem faced by the trustor offer the potential for selecting a less risky action. As instantiated in Table 8.4, this version of the prisoner's dilemma does not offer a less risky option. Note, however, that by changing one of the trustor outcomes, say 8 to 9, and the algorithm's constants to $\varepsilon_1 = 8, \varepsilon_2 = 9$ the situation does then demand situational trust. Overall, the prisoner's dilemma is a borderline case in which the specific values of the outcomes determine whether or not the situation demands trust.

5. The Chicken situation is a prototypical social situation encountered by people. In this situation each interacting individual chooses between safe actions with intermediate outcomes or more risky actions with more middling outcomes. An example might be the negotiation of a contract for a home or some other purchase. This situation, like the Trust Situation, demands trust because it follows the same pattern of risks as the Trust Situation.

Table 8.4 and the analysis that followed examined several situations and employed our conditions for situational trust. In several situations our algorithm indicated that the conditions for trust were met. In others, it indicated that these conditions were not met. We related these situations back to interpersonal situations commonly encountered by people, trying to highlight the qualitative reasons that our conditions match situations involving people. Overall, this analysis provides preliminary evidence that our algorithm does select many of the same situations for trust that are selected by people. While much more psychologically valid evidence will be required to strongly confirm this hypothesis, the evidence in this section provides some support for our hypothesis. We now move on to the problem of measuring trust.

8.2 Measuring Trust

Several trust researchers have recognized the importance of risk in defining, characterizing, and quantifying trust (Deutsch, 1962; Luhmann, 1979, 1990). Risk is typically quantified as the expectation of a loss function (Risk, 2007). Formally,

$$R(x, y) = \sum_i L(x, y)p(y)$$

is the risk associated with predicting x when the true value is y .

Here, we define the loss function to be the difference in outcome associated with a partner's choice of one action over another. In other words, the loss L for individual i

when action a_2^{-i} is selected by individual $-i$ over action a_1^{-i} is equal to,

$$L^i(a_1^{-i}, a_2^{-i}) = |O^i(a_1^i, a_1^{-i}) - O^i(a_1^i, a_2^{-i})| = |_{11}o^i -_{21}o^i|.$$

The expectation of the loss function is $p(a_2^{-i})$ where $p(\cdot)$ is the probability. This expectation can also be conditioned on external evidence, such as the situation's correspondence, the partner's recent history,

etc. Overall then the risk to the trustor associated with selecting the trusting action a_1^i is measured as $R^i(a_1^{-i}, a_2^{-i}) = L^i(a_1^{-i}, a_2^{-i}) \left(p(a_2^{-i}) \right)$. We propose that trust, τ , is proportional to the risk assumed by the trustor given that the situation requires trust, namely $\tau \propto R^i(a_1^{-i}, a_2^{-i})$. Box 8.2 provides an algorithm for measuring trust.

Measuring Trust

Input: Outcome matrix O
Output: Real number τ measuring trust or NULL

1. Use the algorithm from Box 8.1 to generate Boolean b .
2. **If** ($b = \text{false}$) //Determine if the situation requires trust
Return NULL
3. **If** $p^1(a_1^{-i}) \leq k$ //Ensure that the trustor holds belief that
Return NULL //the trustee will select the trusting action
- Else**
Set $\tau = R^i(a_1^{-i}, a_2^{-i}) = L^i(a_1^{-i}, a_2^{-i}) \left(p(a_2^{-i}) \right)$
4. **Return** τ //return the measure for trust

Box 8.2 The algorithm above depicts a method for measuring the trusted required by a social situation.

Using a situation described in section 2.3.2, the investor-trustee game, we successively created situations that placed the trustor (investor) at risk to explore the algorithm's predictions. Recall that the investor-trustee game is a situation that has been used by scientists to explore the neuroscientific origins of trust (King-Casas et al., 2005). The game appoints one individual as the investor (the trustor for this discussion) and other individual as the trustee. The investor is given some quantity of money. He or she chooses some amount to invest with the trustee. The amount invested is multiplied by a factor. Finally the trustee decides how much to give back to the investor. Change in trust has been shown to correlate with investor reciprocity in this game.

In Table 8.5 we demonstrate our algorithm for measuring trust via the investor-trustee game. In our simplified version of the game, the investor is given a quantity of money

delineated in the first column of Table 8.5. The investor has a binary decision choosing to either invest all or half of the money with the trustee. The trustee also chooses between two potential actions, either returning all or none of the appreciated money back to the investor. Money invested with the trustee doubles in value.

Table 8.5 The table demonstrates the change in trust measure with respect to the changing conditions of the investor-trustee game. The table shows that as the probability that the trustee will violate the trust increases, so to does the trust measure. Hence the amount of trust necessary to selected the trusting action increases. Moreover, as the loss increases in relation to the initial money given to the investor the trust measure increases.

Investor-Trustee Demonstration of Algorithm for Measuring Trust													
Initial Investor Money	Outcome Matrix	Situational Trust?	$p(a_2^{-i})$	$L_1^i(a_1^{-i}, a_2^{-i})$	τ								
50	<table border="1"> <tr><td>100</td><td>75</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>25</td></tr> </table>	100	75	0	0	0	25	true	0.0	100	0		
100	75												
0	0												
0	25												
50	<table border="1"> <tr><td>100</td><td>75</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>25</td></tr> <tr><td>100</td><td>50</td></tr> </table>	100	75	0	0	0	25	100	50	true	0.5	100	50
100	75												
0	0												
0	25												
100	50												
50	<table border="1"> <tr><td>100</td><td>75</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>25</td></tr> </table>	100	75	0	0	0	25	true	0.9	100	90		
100	75												
0	0												
0	25												
0	<table border="1"> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td></tr> </table>	0	0	0	0	0	0	false	0.5	0	null		
0	0												
0	0												
0	0												
25	<table border="1"> <tr><td>50</td><td>37.5</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>12.5</td></tr> <tr><td>50</td><td>25</td></tr> </table>	50	37.5	0	0	0	12.5	50	25	true	0.5	50	25
50	37.5												
0	0												
0	12.5												
50	25												
50	<table border="1"> <tr><td>100</td><td>75</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>25</td></tr> <tr><td>100</td><td>50</td></tr> </table>	100	75	0	0	0	25	100	50	true	0.5	100	50
100	75												
0	0												
0	25												
100	50												

The second column of Table 8.5 depicts the outcome matrix resulting from an initial investor amount depicted in the first column. The next column denotes whether or not the situation meets the conditions for situational trust. Only the situation without initial investment fails to meet the conditions for situational trust. In the top half of the table, the

forth column of the table varies the probability that the trustee will choose to not return the investor's investment. When the probability that the trustee will violate the trust is zero (last column), the situation presents no risk. Our measure of trust returns zero, reflecting the risk to the investor. As the probability that the trustee will violate the trust increases so to does our trust measure. The final three rows in Table 8.5 demonstrate the trust measure's change with increasing initial investment.

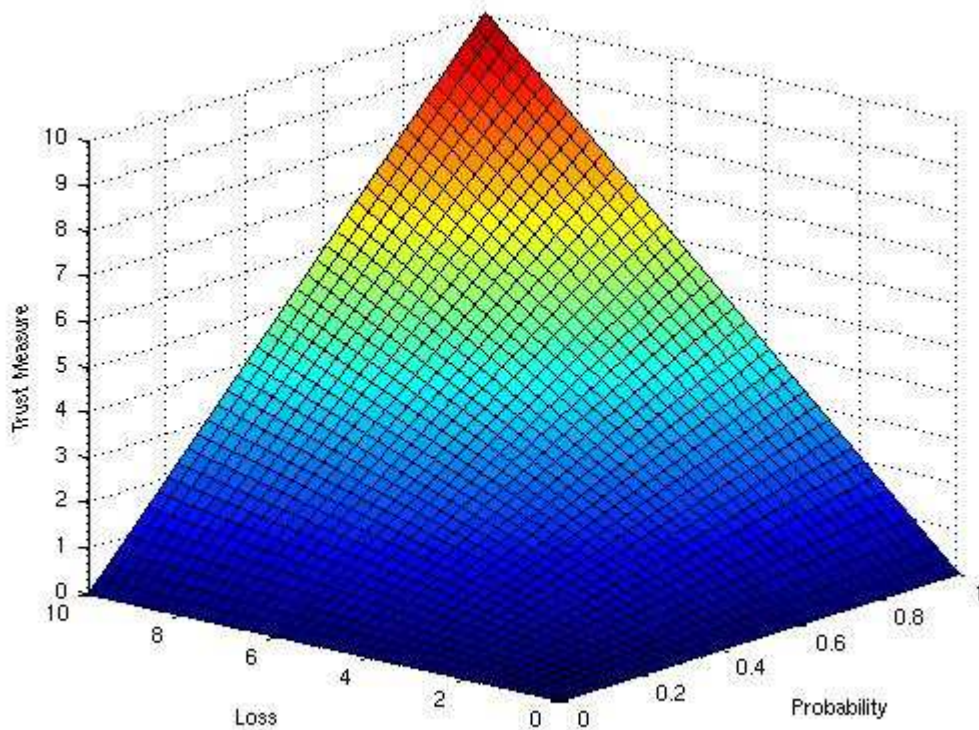


Figure 8.5 Graphical depiction of the increase of our proposed trust measure with respect to increasing loss and probability of untrusting action selection. Our trust measure is a unitless measure which is proportional to the amount of loss and the probability of selecting the untrusting action. The measure is useful for comparing situations that require trust.

Table 8.5 demonstrates our measure of trust for a particular situation. Because the investor-trustee game has typically been tied to money the situation is a good intuitive demonstration that allows us to show various levels of loss and probabilities of loss. In general, the risk associated with the trust measure obeys the function depicted in Figure

8.5. As described, our trust measure is linear with respect to amount of loss and probability of loss. Results from Cumulative Prospect Theory contradict the linearity of our trust measure at extreme amounts of loss (Tversky & Kahneman, 1992). Future refinements of our trust measure could include Cumulative Prospect Theory's measures of expected utility. Regardless, further experimentation involving the use of human subjects will be necessary in order to provide additional evidence that our algorithm for measuring trust is quantitatively accurate.

8.3 Recognizing Relationships that afford Trust

In this final section we connect much of the preceding theory and discussion by asking if a robot can determine if a particular partner can be trusted in a particular situation. The examination of this topic forces the robot to assume the role of the trustor, deliberating with respect to the actions of its human partner. This question is potentially relevant to many robotics problems today. Automatic pilots helping to fly planes and drive trains might question the authority and decision making of the human while in transit. Autonomous robots operating in dangerous locations such as space might reject the actions of a human if they are deemed to put the entire team or mission at risk. While the applications of today do not require robots capable of rejecting the advice or actions of a human, the applications of tomorrow will.

Selecting the most trusted partner requires that a robot have models of all of the potential partners. Hence the robot will need to interact with each partner, constructing models of each individual. Next, the robot faces a situation requiring trust and must select one individual to be the trustee. In this final section, we propose a method for recognizing relationships that afford trust.

8.3.1 Selecting a Trusted Partner

The selection of a trusted partner begins with interaction. The robot must interact and construct a model of all of its potential partners. Once these models have been developed, the robot can then predict each partner's likely action based on knowledge of the partner. Questions of trust are primarily concerned with the trustee's internal tendency to act in a manner that mitigates the risk assumed by the trustor. Thus, to determine that a partner is trustworthy, the robot must conclude that, given some situation \tilde{O} requiring trust, the partner's transformation, θ^{-i} , will be such to create an effective situation O_E in which the partner will select action a_1^{-i} resulting in outcome ${}_{11}o^i > {}_{x2}o^i > {}_{21}o^i$ for the trustor. To make this conclusion the robot must hold the belief that $p^i(a_1^{-i}|\theta^{-i}) > k$. To select the most trusted partner for the situation the robot solves the equation $\arg \max_{-i} p^i(a_1^{-i}|\theta^{-i})$, determining the partner with maximum likelihood of selecting action a_1^{-i} given the partner's transformation. Box 8.3 describes the process algorithmically. The following section proposes a means for evaluating our method of selecting a trusting partner.

Selecting a Trusted Partner

Input: Partners $-i_1, \dots, -i_k$, trusting situation \tilde{O} .

Output: Most trusted partner i^* .

1. The robot interacts with individuals $-i_1, \dots, -i_k$ constructing models $m_1^{-i}, m_2^{-i}, \dots, m_k^{-i}$ for each partner.
2. The robot is then presented with a situation \tilde{O} requiring trust.
3. For each partner the robot generates belief $p^i(a_1^{-i} | \theta^{-i})$
4. The robot selects the most trusted partner by solving $\arg \max_{-i} p^i(a_1^{-i} | \theta^{-i}) = i^*$.
5. **Return** the most trusted partner i^* .

Box 8.3 A method for selecting the most trusted partner among several potential partners is presented.

Consider, for example, coworkers at a dangerous job such as a prison. Both individuals must place their own safety at risk in order to perform tasks, such as checking on inmates. Each individual must believe that if they are attacked while performing a task that the other individual will act in a manner that will mitigate their risks. The trustor in this example is the individual walking and observing inmates. Their outcomes are at risk because they are alone or outnumbered by dangerous people. Their coworker, in the meantime, remains safe but must be able and willing to react and come to their rescue if an attack occurs. The coworker is thus the trustee.

While this situation clearly demands trust, we can now consider how the situation changes if we allow the trustor to interact with and build models of different coworkers. Once the trustor does this we offer them the opportunity to select the coworker whom they trust the most.

8.3.2 Selecting the most Trusted Partner

Table 8.6 Summary of the selecting the most trusted partner experiment.

Experiment Summary	
Selecting the most trusted partner	
Purpose	Investigate the possibility of using the algorithms presented in this chapter to select the most trusted partner.
Experiment Type	Laboratory experiment
Hypothesis	Use of the selecting a most trusted partner algorithm will result in greater outcome obtainment than use of a <i>max_own</i> strategy which does not consider the partner when selecting an action.
Procedure	Follow the procedure presented in Table 8.7.
Independent variable	Experimental versus control condition.
Dependent variable	Average outcome obtained.
Method of Analysis	Ablation experiment consisting of comparison of the experimental condition involving use of the selecting a most trusted partner algorithm to a control condition.
Conclusion	Hypothesis is supported. The average outcome obtained in the experimental condition was significantly greater than the outcome obtained in the control condition.

We conducted a robot experiment to explore the effect of selecting the most trusted partner on the robot’s task performance. The experiment was designed to complement the prison guard example presented above. The robot in this case is tasked with guarding one of two types of prisoners: escape threats or riot threats. The robot also has two potential teammates for the task: a fast but weak human partner and a strong but slow human partner. The fast partner is better able to capture escaping convicts while the strong partner is better able to quell riots.

The purpose of the experiment is to demonstrate the algorithm from Box 8.3 for selecting the most trusted partner. Our experiment compares the same robot in the same situation with and without use of the most trusted partner algorithm. Hence this is an ablation experiment. We hypothesize that use of the algorithm would result in greater average outcome. Our independent variable in this experiment is thus an experimental condition in which the robot uses the algorithm versus control condition in which the

robot does not use the algorithm. In the control condition the robot used a *max_own* strategy to select the action that maximized its own outcome without regard to the partner. The dependent variable here is average outcome obtained by the robot. Outcome obtained is a crude measure of task performance. Table 8.7 summarizes the experiment.

The design of the experiment is meant to notionally resemble a scenario in which selecting the most trusted partner might be critical for the robot's performance and well-being. The scenario chosen focuses on a prison environment in which the robot must select the most trusted partner for a dangerous task. Numerous other scenarios are also possible. This scenario was chosen because it once again highlights the generality of this framework in different environments.

Experimental Setup

The experiment was conducted in a laboratory environment. Figure 8.6 depicts the layout. Two notional prison cells are located next to one another with a divider preventing the robot from observing both cells at one location. The robot's base is located approximately seven feet from both cells in a straight line. The top half of Figure 8.6 (left-most photo) depicts the robot at its base position. A human operator sat at a notional prison operations desk. The robot's base is not located within the sight of the human operator.

The robot used gestures to communicate its partner preference to the human operator. Notionally, the human operator would then assign a human teammate to the robot. To make its partner preference known the robot moved forward approximately three feet into an area easily observed by the operator.

Experimental Setup

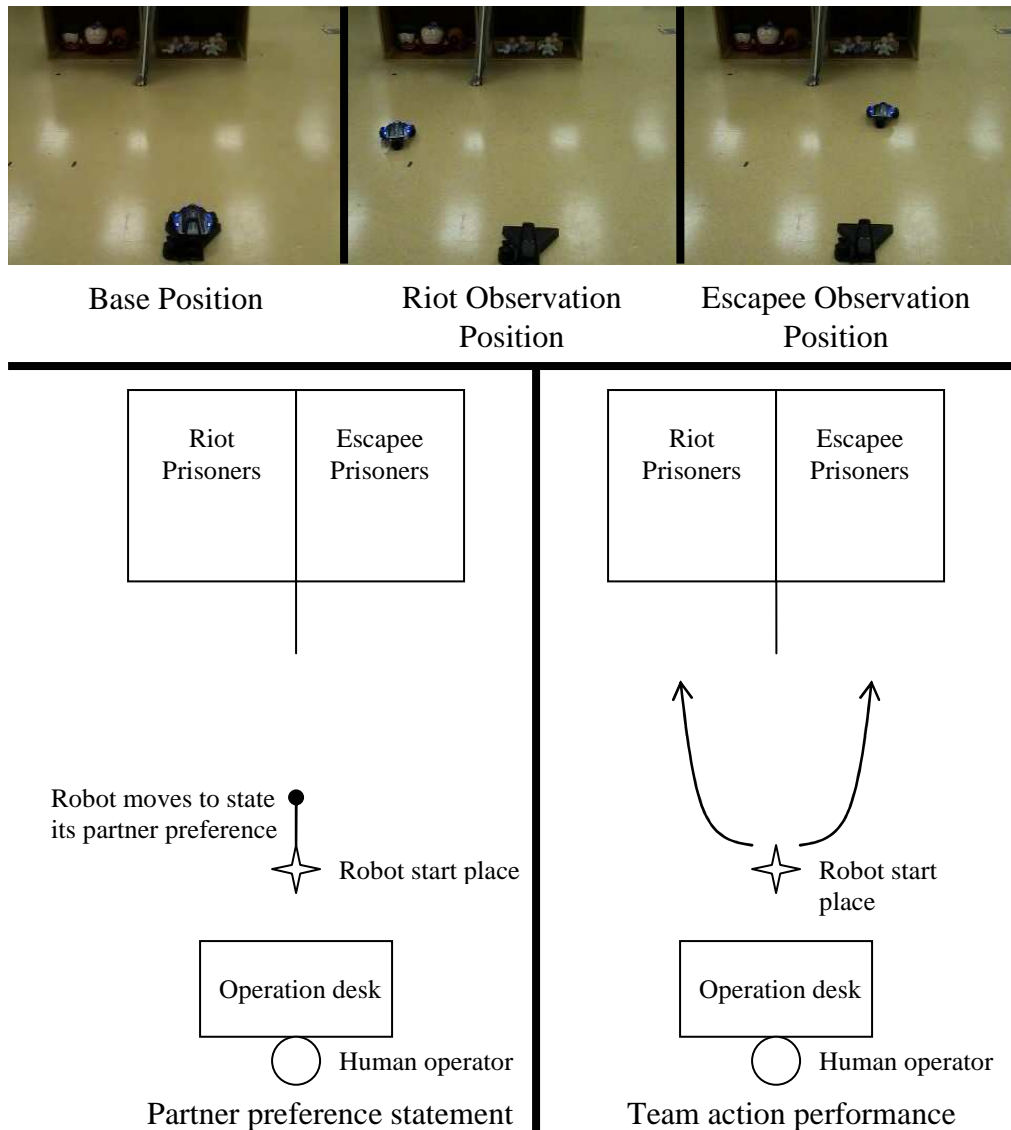


Figure 8.6 The top of the diagram shows the laboratory setup for the most trusted partner experiment. Left photo shows the base position which is located about 10 feet in front of two containers representing cell blocks. The center position shows the robot at an observation position in front of riot prisoners. The right photo depicts the robot observing the escapee prisoners. The two lower diagrams depict the actions the robot performs in the experiment. In the left diagram the robot first moves to a position within view of the operator and then moves to state “yes” or “no” with respect to its partner preference. In the right diagram, the robot moves to observe either the riot prisoners or the escapee prisoners.

The robot’s convict observation actions were performed by moving to locations in front of the two cells. The center image in the top half of Figure 8.6 depicts the robot observing the cell containing the riot threat convicts and rightmost image depicts the robot observing the cell containing the escape threat convicts. The bottom half of Figure

8.6 portrays the robot's motions when stating its partner preference and making its prison cell observations.

Partner Preference Statement

When the robot selects its partner it must communicate its selection to the human operator. The process of partner selection has three stages: 1) the human operator asks the robot in a verbal statement if it prefers partner x ; 2) the robot uses the algorithm in Box 8.3 to determine which partner it prefers; 3) if the partner preferred by the robot is the same as the one asked about by the human operator then the robot produces a "yes" motion, if not then the robot produces a "no" motion. Figure 8.7 depicts the robot stating "yes" and Figure 8.8 depicts the robot stating "no." To state "yes" the robot moved its camera neck in an up-and-down motion imitating the same motion a human makes when shaking their head yes. To indicate "no" the robot turned back and forth through approximately 180 degrees imitating a human shaking their head no.

Rovio indicating Yes

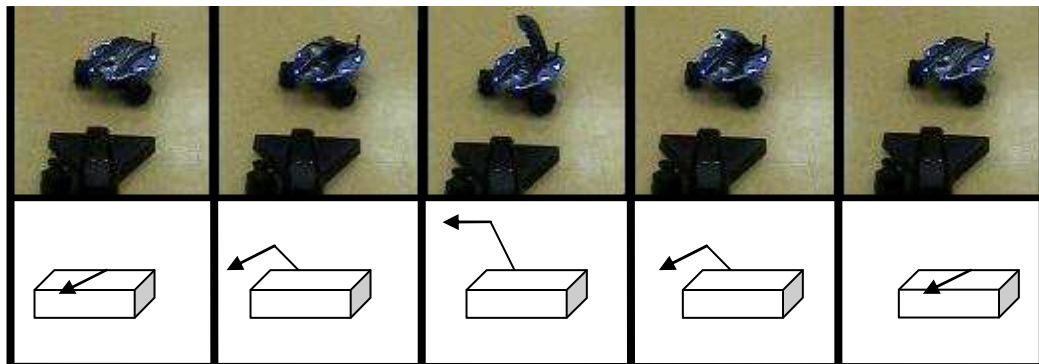


Figure 8.7 Robot movement for stating "yes" to the operator's question regarding its partner preference. The robot moves its neck up and down to state yes.

Rovio indicating No

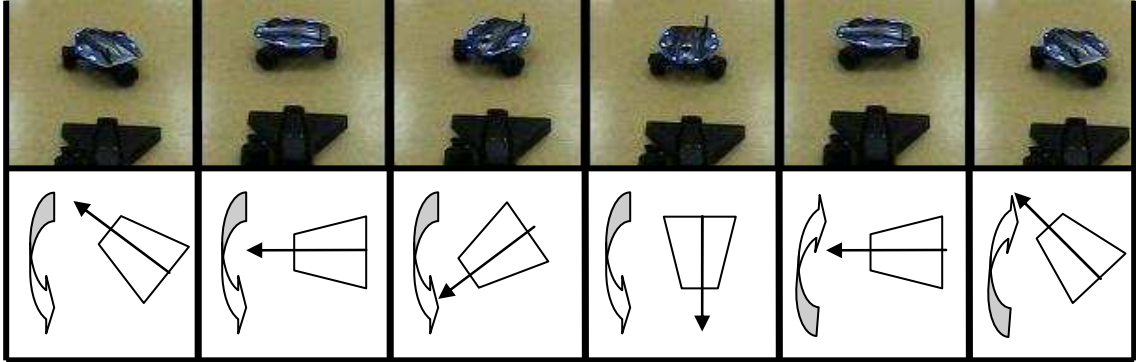


Figure 8.8 Robot movement for stating “no” to the operator’s question regarding its partner preference. The robot moves back and forth in a half circle to indicate no.

Partner Models

Prior to the experiment two models of human partners were created. Both models contained the actions: `chase-convict`, `ignore-escape`, `defend-from-attack` and `flee-riot`. The `chase-convict` action results in the human partner chasing an escaping convict. The `flee-riot` action results in the human partner fleeing the prison. The `defend-from-attack` action results in the human partner defending the robot and the prison during an attack. The `ignore-escape` action results in no action from the human partner. The robot’s model of the **fast** human partner contained the belief $p(a^{-i})$ that that human would select the `chase-convict` action with a probability of 0.90, the `ignore-escape` action with a probability of 0.10, the `defend-from-attack` action with a probability of 0.10, and `flee-riot` action with a probability of 0.90. The robot’s model of the **strong** human partner contained the belief $p(a^{-i})$ that that human would select the `chase-convict` action with a probability of 0.10, the `ignore-escape` action with a probability of 0.90, the `defend-from-attack` action with a probability of 0.90, and `flee-riot` action with a probability of 0.10. Hence, the robot’s models of the two

partners' indicated different action preferences. Again these models were given to the robot.

Experimental Procedure

In addition to stating its partner preference the robot's action model consisted of the following two actions: `observe-riot-convict` and `observe-escapee-convict`. The `observe-riot-convict` resulted in the robot moving to a position where it could observe the cell containing the convicts with a potential to riot. The `observe-escapee-convict` resulted in the robot moving to a position where it could observe the cell containing the convicts with a potential to escape.

Prior to experimentation, twenty outcome matrices meeting the conditions for trust were created. These twenty outcome matrices were used in both the experimental and control conditions. As mentioned above, the robot's partner models were constructed prior to experimentation by the experimenter. Hence, use of the algorithm for selecting most trusted partner began with step two of Box 8.3. In this step, the robot is presented with one of the twenty outcome matrices created prior to experimentation, \tilde{O} . The experimental procedure is listed below:

Table 8.7 Experimental procedure for the most trusted partner experiment.

Experimental Procedure	
1)	Twenty outcome matrices populated with random values (arbitrary range of $[-20,20]$ was used) meeting the condition for trust (Box 8.1) were created.
2)	The robot is presented with a new outcome matrix \tilde{O} (one of the twenty created in step 1). The robot is randomly tasked with observing either the convicts with a potential for rioting or the convicts with a potential for escaping.
3)	Experimental condition: The algorithm from Box 8.1 is used to determine if the situation demands trust.

- 4) **Experimental condition:** For each of the two potential partners, the robot uses the partner models to retrieve belief $p(a_1^{-i})$ that the partner will select the trusted action.
- 5) **Experimental condition:** The robot selects the partner with the greatest likelihood of selecting the most trusted action.
- 6) **Experimental condition:** The human operator verbally asks the robot if it would prefer to have one of the two potential partners as a teammate.
- 7) **Experimental condition:** If the robot is asked to be a teammate with the same partner that it prefers then it makes the yes motion, otherwise it makes the no motion.
- 8) **Experimental condition:** The robot selected the trusting action.
Control condition: The robot selected the action that maximized its own outcome without regard to the partner (*max_own*).
- 9) The robot moves to either observe the convicts with a potential for rioting or convicts with a potential for escaping (Figure 8.9).
- 10) Notionally, the convicts with a potential for rioting attempt to riot and the convicts with a potential for escaping attempt to escape.
- 11) The robot observes the convict's actions and the human teammate selects an action according to its action preference relation.
- 12) The robot receives maximal outcome (actual value depends on \tilde{O}) if the convicts attempting to escape are captured and the convicts attempting to riot are prevented from rioting. This occurs if the robot has selected the strong human partner as a teammate when it is observing convicts with the potential for rioting and if the human partner selects the defend-from-attack action, which it does with probability of 0.90. The robot also receives maximal outcome if it has selected the fast human partner as a teammate when it is observing convicts with the potential for escaping and if the human partner selects the chase-convict action, which it does with probability of 0.90. All other combinations of robot and human action result in reduced outcome (actual value depends on \tilde{O}).

In essence, the experimental condition used our algorithm for selecting the most trusted partner whereas the control condition did not.

The Robot's Observation Actions



Convicts with a potential
to riot

Convicts with a potential
to riot

Convicts with a potential
to riot in the dark

Figure 8.9 Examples of the robot's observation actions from the two (left and center) prisoner observation points. The image to the right depicts an experimental trial conducted under limited lighting.

Results

We ran twenty trials in each of two conditions: one control condition and one experimental condition. Recall that the purpose of the experiment is to show that use of our method for selecting the most trusted partner by a robot results in greater average outcome in situations demanding trust.

As depicted in Figure 8.10 the average outcome received in the control condition was -7.24 versus 10.57 in the experimental condition in which the robot used our algorithm to select the most trusted partner. This difference was statistically significant ($p < 0.03$). In terms of partner selection, in the experimental condition the robot consistently selected the best partner. In the control condition, on the other hand, the robot selected the best partner in 35 percent of the trials. With respect to the results for the team, in the experimental condition the human robot team received an average of 13.05 outcome per trial versus -2.19 for the control condition. This difference however was not statistically significant ($p \sim 10.1$).

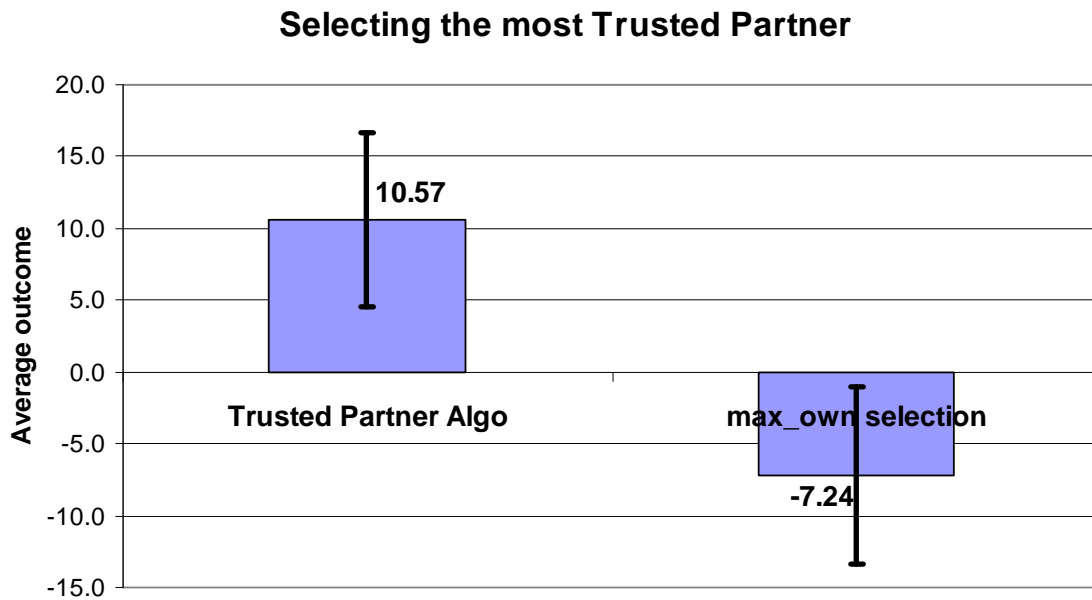


Figure 8.10 Results from the selecting the most trusted partner experiment. When the robot uses the algorithm from Box 8.3 to select the most trusted partner the average outcome was 10.57. The control condition, in contrast, in which the robot used a *max_own* strategy to select its action without consideration of the partner resulted in an average outcome of -7.24. The difference between these two conditions was statistically significant ($p < 0.03$).

The results support our hypothesis that our algorithm for selecting the most trusted partner does indeed aid the robot's task performance. It is not clear, however, that the use of the algorithm aids team performance. Although the team results indicate a difference, the difference was not found to be statistically significant. The fact that we used only twenty trials in each condition is likely the cause for this lack of significance. Still, the team results indicate that selection of the most trusted partner by the trustor does not always aid the trustee. This again demonstrates the one-way nature of trust. Namely, that acts of risk mitigation by the trustee may not have a positive impact on the trustee's outcomes.

This experiment, in and of itself, serves as a demonstration of our algorithm for selecting the most trusted partner which is based on the theoretical principles described in sections 8.1 through 8.3. It does not, however, serve as a conclusive proof of the

algorithm we have presented. This experiment is meant to complement the other experiments by demonstrating a particular situation in which a real robot selects actual actions as part of a grounded situation. Further testing in additional domains will be necessary to show that the algorithm and the results apply to other domains.

8.4 Conclusions: Trust in Human-Robot Interactions

The preceding chapter has introduced ideas for defining, representing, and measuring trust based on interdependence theory. We have presented algorithms for segregating those social situations that demand trust from those which do not, algorithms for measuring trust, and algorithms for selecting the most trusted partner from several potential candidates.

Many different types of experiments have been presented in this chapter. Some of the experiments have consisted of simple demonstrations, numerical simulations, and also laboratory experiments involving the use of real robots. Our results have both supported our hypotheses and refuted them. Overall, the research and the results presented serve more as an introduction to the approaches presented herein than as a conclusion. These ideas will need to be further tested on a variety of hardware systems and in a multitude of environments. We believe that these ideas will serve as a basis for various different research avenues. Moreover, although the evidence for the theories and hypotheses presented may not be as complete as desired, it is the breadth and scope of our framework that offers the most potential for robotics and for artificial intelligence.

CHAPTER 9

CONCLUSIONS

In this final chapter we summarize the principal results of our research, discuss directions for future work, and present some final remarks.

9.1 Summary of Contributions

This dissertation makes the following contributions:

- **A general, computational framework implemented on a robot for representing and reasoning about social situations and interaction based on interdependence theory.** We have presented a computational framework based on interdependence theory that affords a means of representing interactions and social situations as outcome matrices. As argued in Chapter 5, outcome matrices are an established method for representing interaction in game theory, experimental economics, and neuroscience communities (Kelley, 1979; Osborne & Rubinstein, 1994). Moreover, the presented framework is general in the sense that the results that have been presented are not tied to a particular robotic system, social situation, environment, or type of human partner.
- **A principled means for classifying social situations that demand trust on the part of a robot and for measuring the trust required by a situation in which a robot interacts with a human.** This dissertation has introduced a novel, general, and principled method for representing and reasoning about trust. Using a definition for trust developed from a lengthy literature review, a series of

conditions for trust have been expounded. In Chapter 8 we showed that these conditions could be used to determine if a particular interaction demands trust on the part of the robot or the robot's partner. Further, we have argued that the amount of trust can be measured as risk and developed methods for measuring trust.

- **A methodology for investigating the theory underlying human-robot interaction.** This dissertation has explored a top-down methodology for exploring the theory that underlies human-robot interaction. This top-down methodology begins with the definition of concepts such as relationship and trust (Chapters 7 and 8). These concepts are then related to our computational representation of interaction and social situations. We then develop general purpose algorithms that tie these concepts to the robot's interactions. This top-down methodology stands in contrast to the bottom-up methodologies typical in most current day human-robot research (Fong, Nourbakhsh, & Dautenhahn, 2003).
- **A computational framework for social action selection implemented on a robot.** Interdependence theory postulates the existence of a process for social action selection that includes a person's own internal predispositions. This process is called the transformation process. This dissertation has demonstrated that the transformation process can be used by a robot for social action selection (section 7.3).
- **An algorithm that allows a robot to analyze and characterize social situations.** This dissertation has presented an algorithm (Box 6.1) that allows a robot to map an interaction to a portion of the interdependence space and, by

doing so, characterize the situation in terms of its interdependence properties (section 6.1). This characterization affords information that has been shown to be an important factor for social action selection and increased task performance (as measured by outcome obtainment).

- **Methods for modeling the robot’s human partner and for characterizing a robot’s relationship with the partner.** This dissertation has explored and developed techniques that afford a robot the ability to model its human partner (section 5.3). Moreover, we have shown that these techniques can be extended to learning about clusters of human types, or stereotype learning (section 5.4). We have shown that stereotyped models of human partners can bootstrap the process of learning about a particular partner. Finally, we have demonstrated techniques by which a robot can determine its partner’s underlying type (section 7.3).

9.2 Research Questions Revisited

The first chapter detailed several questions that we intended to explore. In this section we review these questions stating the conclusions this work has set forth.

- 1) **What effect will the development of a theoretical framework that allows a robot to represent social situations and recognizing situations that require trust have on the robot’s ability to select actions?**

The development of a theoretical framework allowing a robot to represent and recognize situations that require trust has been shown to afford better partner selection and task performance (sections 8.1 through 8.3). Moreover, this dissertation has shown that the creation of the theoretical framework, in and of itself, allows for a general and principled investigation of human-robot interaction

(section 4.1). Finally, the representations for interaction and social situations described and developed as a part of this research have played an integral role in the creation of methods for everything from stereotype learning (section 5.4) to reasoning about trust (section 8.1).

2) What effect will deliberation with respect to a social situation have on the robot's ability to select actions?

This research has resulted in algorithms that allow the robot to deliberate with respect to the robot's human partner (section 5.3), the interdependence characteristics of the situation (section 6.1), and the dyad's relational disposition (section 7.1). In all of these cases, the robot's consideration of the different aspects of the interaction has resulted in greater outcome obtainment which is a general reflection of better task performance.

3) What effect will algorithms, developed as part of the theoretical framework of social situations, that allow a robot to represent its relationship with its human partner and to characterize these relationships in terms of the trust have on the robot's ability to select actions?

This dissertation has presented algorithms that allow a robot to reason about the relational disposition of its partner (section 7.1). Moreover, we have demonstrated that the methods and techniques created pertaining to trust offer the robot a means for selecting the best partner for a task (section 8.3). Hence, these results serve as evidence that a robot's ability to represent its relationships and to characterize these relationships in terms of trust is affords the robot techniques for improved task performance as determined by the outcome it obtains.

Principal Research Question

4) What effect will characterizing the trustworthiness of social relationships and of social situations have on a robot's ability to select actions?

Overall, the characterization a robot's social relationships and situations in terms of trust has an important effect on the robot's social performance. As demonstrated by sections 5.3, 5.4, 6.2, and 8.3 this characterization affords improved task performance. In a larger sense, the ability of the robot to represent and reason about trust, relationships, and the interdependence characteristics of the situations that it faces has a significant impact on the robot's social behavior. As indicated by the results from sections 5.3, 5.4, 6.2, 7.3 and 8.3, a robot capable of deliberating about its interactions, its partner, and the social situation, is better suited to act in an appropriate manner in a wider variety of situations than a robot which lacks these capabilities.

9.3 The Road Ahead

The presented framework offers numerous avenues for potential research. Applications of this work could conceivably touch many different areas of artificial intelligence. Some of the most promising and immediate applications of this work is in the domain of assistive therapy. Assistive therapy often involves one-on-one interactions with the same person over the course of the treatment, a good opportunity for partner modeling (Feil-Seifer, 2008). Moreover, therapeutic treatments often involve repeatedly asking the patient for their current state. This information could potentially be translated in outcomes.

Before discussing long-term avenues of this research, we will briefly describe how this framework could be deployed on a fielded robotics system. On a deployed system it will be necessary for the robot to generate outcome values reflecting the partner's state.

Smile/frown detectors, pose detectors, and affect in speech recognition could all potentially be used to provide estimates of the partner's outcome value. The robot's outcome value, on the other hand, will likely be tied to task performance for most applications. Embuing the robot with action recognition presents is a significant challenge to deploying this framework. Nevertheless, researchers have begun to explore this challenge (Philipose et al., 2004; Picard, 2000). The implementation of these underpinnings should allow a robot to create outcome matrices representing its interactions. Sensor noise and uncertainty present additional challenges. Game theory offers a variety of techniques for managing uncertainty which could potentially be explored to address these challenges (Osborne & Rubinstein, 1994). Once the preceding concerns have been addressed, the framework should be capable of interactive action selection on a fielded robotics system.

Although the development of fielded systems is an important next step for this research, it is the theoretical extensions of this framework that hold the most promise to advance our understanding of human-robot interaction.

- **Emotion** – The outcome values described throughout this dissertation may serve as a placeholder for emotion. Emotion is an important and active area of research within the artificial intelligence community (Velasquez & Maes, 1997). The relation of outcome values to emotion is unclear, yet certainly important. Outcome values are defined as scalar real numbers, but emotions are often described multi-dimensionally (Ortony, Clore, & Collins, 1988). Do outcome values serve as a scalar descriptor of emotion? Or can interdependence theory be used to explain emotions? Fear, for example, could be described as the negative outcomes resulting from deliberation with respect to future negative outcomes. Jealousy could similarly be delineated as negative outcomes resulting from positive outcomes received by another individual. Still, other emotions, such as anger, seem difficult to describe as an outcome value. A

framework tying the broad areas of emotion and interdependence together would be an important theoretical result for both artificial intelligence and psychology.

- **Symbol Grounding** – Symbol grounding refers to the problem of how seemingly meaningless symbols are translated into meaningful monikers for artifacts in the real world (Harnad, 1990). The framework detailed in this dissertation offers the possibility of grounding symbols in terms of the reward and cost they afford the robot. For example, the symbol of firefighter comes to represent the actions and outcomes afforded by a firefighter in specific situations. When and if this framework becomes tied to emotion, it may then be possible to describe symbols such as firefighter in terms of the emotion that the symbol produces. This connection would begin to touch Damasio’s somatic-marker hypothesis, which states that emotional processes guide behavior via associations with emotion imprinted memories (Damasio, 1994).
- **Deception** – Deception is generally defined as “causing another to believe what is not true; to mislead or ensnare (Deception, 1999)”. McCleskey notes that deception is a deliberate action or series of actions brought about for a specific purpose (McCleskey, 1991). The framework presented in this dissertation offers a means of understanding and reasoning about deception. Deception can be modeled as an action or series of actions taken by the deceiver with the purpose of influencing the target to select a particular action or series of actions. The question then becomes how the robot’s model of the partner influences its ability to deceive.
- **A general theory of interaction** – Many of the concepts and ideas discussed in this dissertation relate not only to robots but to any system of interacting entities. Whether one is exploring how two companies interact to maximize profit and cooperation, a husband and wife interact to minimize fighting, or a human and a machine interact to perform a task, many of the same principles apply. It may be possible to forge a

general theory of interaction encompassing all of these seemingly disparate fields, and in doing so, provide tools for each field to move forward.

9.4 Towards a Socially Intelligent Robot

While the dream of creating a sociable robot is still a great many years away, the theory, and principles on which these social beings will be based must be created today. This dissertation has approached this challenge by extending and adapting theories of human social psychology and game theory to the problems faced by an interacting human and robot. The theories and principles developed herein have been formulated from first principles and generally accepted definitions. From these definitions we have crafted and tested algorithms. We have shown that this approach leads to research that is not tied to a particular robotic platform, environment, or human. We feel that this type of underlying scientific theory will be critical for the future success of the human-robot interaction field.

The challenge of creating sociable robots uniquely bridges human psychology and artificial intelligence. Simple optimization algorithms are unlikely to succeed in a way that results in naturalistic interaction. Moreover, it will not be feasible to perfect a robot's interactions. The imperfection of human socialization plays a large role in defining us as humans. The interactive, socially intelligent robots of the future should share our social fallibility (Sharkey & Sharkey, 2010 in press). Imagine the horror of interacting with a robot that has optimized for every rebuttal, every joke, every tender moment. Social intelligence is defined by a human's social flexibility in a myriad of different situations and different partners. The creation of a new socially intelligent being will likely tell us as much about ourselves as it will tell us about robotics.

APPENDIX A

GLOSSARY OF TERMS

Actor script: A predefined set of interactive instructions that an individual follows when interacting with the robot. Used to control the human's behavior.

Basis of control: The ways in which each partner affects the other's outcomes (Kelley & Thibaut, 1978).

Belief: A possibly uncertain truth statement held by an individual.

Bilateral Actor Control (BAC): The human or robot's ability to affect its own outcomes in a social situation (Horswill, 1998; Kelley & Thibaut, 1978).

Concurrent interaction style: A style of interaction in which both individuals select actions at the same time.

Correspondence: The extent that each partner's outcomes are consistent with the others (Kelley & Thibaut, 1978).

Diagnostic situation: A situation or network of situations that is used by an individual to assign credit for a partner's action selection to either the partner or the environment (based on Rusbult & Van Lange, 2003).

Disposition: A durative or predominant tendency with respect to an individual's social character.

Dyad: A group of two; a couple; a pair (Dyad, 2006).

Dyadic interaction: One-to-one interaction occurring between only two individuals.

Effective situation: A conceptual term used to denote the cognitively transformed and internal representation of a social environment that an individual uses to determine how to act (Kelley & Thibaut, 1978).

Given situation: A conceptual term used to denote the direct perceived experience of a social environment (Kelley & Thibaut, 1978).

Individual: Either a human or a social robot.

Interaction: influence—verbal, physical or emotional—by one individual on another. (Sears, Peplau, & Taylor, 1991).

Interdependence: The extent that each partner's outcomes are influenced by the other partner's actions (Kelley & Thibaut, 1978).

Interdependence space: A four dimensional space used to represent all dyadic social situations (based on (Kelley et al., 2003).

Mutual Joint Control (MJC): The individual's ability to affect both its own outcomes and the outcomes of its partner in a social situation (Kelley & Thibaut, 1978).

Mutual Partner Control (MPC): The individual's ability to affect their partner's outcomes in a social situation or interaction (Kelley & Thibaut, 1978).

Outcome: A unitless scalar value representing an individual's utility, reward, and/or happiness.

Outcome matrix: A conceptual and computational representation of an interaction and social situation that includes information about the individuals interacting, actions available, and resulting outcome for the selection of an action pair by the dyad (Kelley & Thibaut, 1978).

Outcome matrix deconstruction: The algorithmic process of separating an outcome matrix into the BAC, MPC, and MJC representing three distinct types of control.

Partner feature: Perceptual features related to the recognition and state determination of the partner.

Partner state: The emotional, behavioral, or physical state of the partner.

Partner type: A partner's classification in terms of disposition with respect to a space of types.

Relationship: A particular type of connection existing between individuals related to or having dealings with each other (Relationship, 2000).

Risk: The expectation of a potential loss of outcome.

Stereotype: An interpersonal schema relating perceptual features to distinctive clusters of traits (Sears, Peplau, & Taylor, 1991).

Situation: A particular set of circumstances existing in a particular place or particular time (Situation, 2007).

Situation analysis: A two-step algorithmic process that uses an outcome matrix to produce a situation's location in interdependence space.

Situation feature: Perceptual features related to the recognition and consideration of the situation and/or social environment.

Situation-based interaction: Interaction that includes consideration of the environmental factors or social situation, as well as the interacting individuals themselves, as influences of interactive behavior.

Situation network: A finite state representation of causally connected social situations that is used to describe the movement to and from situations resulting from mutual interactive behavior selection (based on Kelley, 1984).

Social environment: Any environment with more than one social robot or human.

Social learning: Improvement with respect to some performance measure on some class of tasks with experience derived from a **social** environment.

Social situation: the social context surrounding an interaction between individuals (Rusbult & Van Lange, 2003).

Socially deliberative pathway: A computational process by which deliberation over the individual's motives and other internal predilections is included in the individual's action decision.

Socially reactive pathway: A computational process in which deliberation over the individual's motives and other internal predilections is not included in the individual's action decision.

Symmetry: The degree to which the partners are equally dependent on one another (Kelley & Thibaut, 1978).

Transformation: A computational method, applied to the values of an outcome matrix that results in the selection of an action.

Transformation process: The process by which a given situation is modified to include the individual's own internal tendencies and concerns to produce an effective situation (Kelley & Thibaut, 1978).

Trust: A belief, held by the trustor, that the trustee will act in a manner that mitigates the trustor's risk in a situation in which the trustee has put its outcomes at risk.

Trustee: In a social situation meeting the conditions for trust, the individual that must determine whether to act in a manner that alleviates the trustor's risk.

Trustor: In a social situation meeting the conditions for trust, the individual the must decide whether to place their outcome at risk or not.

Turn-taking interaction style: A style of interaction in which individuals iteratively select interactive actions.

Unpopulated outcome matrix: An outcome matrix devoid of outcome values.

Untrusting action: A potential action for the trustor that does not entail risk.

APPENDIX B

EXAMPLE SOCIAL SITUATIONS

The following list describes several canonical social situations. Sixteen situations are presented (Kelley et al., 2003). These situations represent several different areas of the interdependence space. The abbreviations denote the interdependence space location in Figure 6.2. The outcome matrices depicted by are normalized. The situation's potential for meeting the conditions for situational trust is listed.

Social Situations					
Name	Verbal Description (based on Kelley et al., 2003)	Outcome Matrix		Interdependence Space Location	Situational Trust?
				I-space abbr.	
Chicken Situation	Each individual chooses between safe actions with middling outcomes and risky actions with extreme outcomes.	8 8 4 12	12 4 0 0	1.0, 0.2, -0.3, 0.0 CHK	Yes
Competitive Situation	Each individual gains from the other individual's loss. Maximal outcome is gained through non-cooperation.	6 6 0 12	12 0 6 6	0.5, -1.0, -0.5, 0.0 COMP	No
Conflicting Coordination Situation	Each individual's outcomes depend on the other individual, yet both individuals action preferences are in conflict.	12 0 0 12	0 12 12 0	1.0, -1.0, 1.0, 0.0 CNCO	No
Cooperative Situation	Each individual receives maximal outcome by cooperating with the other individual.	12 12 6 6	6 6 0 0	0.5, 1.0, -0.5, 0.0 COOP	No
Correspondent Coordination Situation	Each individual's outcomes depend on the other individual and both individuals action preferences correspond.	12 12 0 0	0 0 12 12	1.0, 1.0, 1.0, 0.0 CRCO	No
Hero Situation	Individuals have a mutual desire to coordinate their actions but a conflict of interest exists as to which action to choose.	15 8 2 4	0 0 5 12	0.7, 0.5, 0.3, 0.1 HERO	Yes
Independent Situation	The action selected by each individual has no impact on the outcome received by the other individual.	12 12 12 0	0 12 0 0	0.0, 0.0, -1.0, 0.0 IND	No

Investor-Trustee Situation	This situation is a trust situation for the investor and a prisoner's dilemma situation for the trustee.	36	23	1.0, -0.3, -0.3, 0.3	Yes
		24	5		
Martyr Situation	Individuals have a weak mutual desire to coordinate their actions but a strong conflict of interest as to which action to choose.	12	18	0.4, 0.0, -0.3, 0.8	Yes
		48	10		
Exchange Situation	Each individual has a choice as to whether or not to have a positive or negative impact on the other individual.	30	20	1.0, 0.0, -1.0, 0.0	No
		8	12		
Prisoner's Dilemma Situation	Both individuals are best off if they act non-cooperatively and their partner acts cooperatively. Cooperation and non-cooperation, results in intermediate outcomes for both.	60	10	0.8, -0.8, -0.6, 0.0	No
		4	0		
Strong Threat Situation	One individual has greater control over the dyad's outcomes. The other individual, if exploited, has significant power to reduce the outcomes of both individuals.	12	12	0.8, 0.0, -0.6, 0.0	No
		12	0		
Trust Situation	In this situation, cooperation is in the best interests of each individual. If, however, one individual suspects that the other will not cooperate, non-cooperation is preferred.	0	0	1.0, 0.2, -0.3, 0.0	Yes
		12	0		
Asymmetric Investor-Trustee Situation	Same as the investor-trustee situation, except that the situation's asymmetry is increased.	8	12	1.0, -0.9, -0.1, -0.7	Yes
		8	0		
Slight Asymmetric Situation	Individual two has slightly greater control over individual one's outcomes. Still, both individuals action preferences correspond.	0	4	1.0, 0.8, 0.2, -0.2	No
		12	4		
Weak Threat Situation	One individual has greater control over the dyad's outcomes. The other individual, if exploited, has limited power to reduce the outcomes of both individuals.	10	14	0.5, 0.0, 0.5, 0.0	No
		13	11		
Investor-Trustee Situation	This situation is a trust situation for the investor and a prisoner's dilemma situation for the trustee.	21	17	Not listed	
		6	5		
Martyr Situation	Individuals have a weak mutual desire to coordinate their actions but a strong conflict of interest as to which action to choose.	7	1	Not listed	
		19	9		
Exchange Situation	Each individual has a choice as to whether or not to have a positive or negative impact on the other individual.	1	10	EXCH	
		7	15		
Prisoner's Dilemma Situation	Both individuals are best off if they act non-cooperatively and their partner acts cooperatively. Cooperation and non-cooperation, results in intermediate outcomes for both.	6	12	PRD	
		12	6		
Strong Threat Situation	One individual has greater control over the dyad's outcomes. The other individual, if exploited, has significant power to reduce the outcomes of both individuals.	0	6	STHR	
		8	0		
Trust Situation	In this situation, cooperation is in the best interests of each individual. If, however, one individual suspects that the other will not cooperate, non-cooperation is preferred.	6	0	TRU	
		8	4		
Asymmetric Investor-Trustee Situation	Same as the investor-trustee situation, except that the situation's asymmetry is increased.	6	12	WTHR	
		12	6		
Slight Asymmetric Situation	Individual two has slightly greater control over individual one's outcomes. Still, both individuals action preferences correspond.	0	6	Not listed	
		7	15		
Weak Threat Situation	One individual has greater control over the dyad's outcomes. The other individual, if exploited, has limited power to reduce the outcomes of both individuals.	6	0	Not listed	
		12	6		

APPENDIX C

LIST OF TRANSFORMATION TYPES

The following table depicts a list of transformation types developed for this dissertation, a description of the robot's character if the robot often selects the transformation type, and the mathematical method for performing the transformation.

Transformation Types		
Name	Character Description	Transformation Method
max_own	Egoistic —the individual selects the action that most favors their own outcomes.	No change
min_own	Ascetic —the individual selects the action that minimizes his/her own outcomes.	${}_{xy}\hat{o}^1 = \max({}_{xy}o^1) - {}_{xy}o^1$
max_other	Altruistic —the individual selects the action that most favors their partner.	${}_{xy}\hat{o}^1 = {}_{xy}o^2$
min_other	Malevolence —the individual selects the action that least favors the partner.	${}_{xy}\hat{o}^1 = \max({}_{xy}o^2) - {}_{xy}o^2$
max_cert	Risk-averse —the individual selects the action that results in the maximal guaranteed outcome.	if $(\min({}_{11}o^1, {}_{21}o^1) = \min({}_{12}o^1, {}_{22}o^1))$ ${}_{xy}\hat{o}^1 = \max({}_{xy}o^1)$ else if $(\min({}_{11}o^1, {}_{21}o^1) > \min({}_{12}o^1, {}_{22}o^1))$ ${}_{x1}\hat{o}^1 = \max({}_{xy}o^1)$ else ${}_{x2}\hat{o}^1 = \max({}_{xy}o^1)$
min_cert	Risk-seeking —the individual selects the action that results in the minimal guaranteed outcome.	if $(\min({}_{11}o^1, {}_{21}o^1) = \min({}_{12}o^1, {}_{22}o^1))$ ${}_{xy}\hat{o}^1 = \max({}_{xy}o^1)$ else if $(\min({}_{11}o^1, {}_{21}o^1) < \min({}_{12}o^1, {}_{22}o^1))$ ${}_{x1}\hat{o}^1 = \max({}_{xy}o^1)$ else ${}_{x2}\hat{o}^1 = \max({}_{xy}o^1)$
max_joint	Cooperative —the individual selects the action that most favors both their own and their partner's interests.	${}_{xy}\hat{o}^1 = {}_{xy}o^1 + {}_{xy}o^2$
min_joint	Vengefulness —the individual selects the action that is most mutually disagreeable.	${}_{xy}\hat{o}^1 = \max({}_{xy}o^1 + {}_{xy}o^2) - ({}_{xy}o^1 + {}_{xy}o^2)$
max_diff	Competitive —the individual selects the action that results in the most relative gain to that of its partner.	${}_{xy}\hat{o}^1 = {}_{xy}o^1 - {}_{xy}o^2 $
min_diff	Fair —the individual acts in a manner that results in the least disparity.	${}_{xy}\hat{o}^1 = \max({}_{xy}o^1 - {}_{xy}o^2) - {}_{xy}o^1 - {}_{xy}o^2 $

REFERENCES

- Abbeel, P., & Ng, A. (2004). Apprenticeship Learning via Inverse Reinforcement Learning. In *Proceedings of International Conference on Machine Learning*, Banff, Canada.
- Altman, I., & Taylor, D. A. (1973). *Social penetration: the development of interpersonal relationships*. New York, NY: Holt, Rinehart, and Winston.
- Angluin, D., & Krikis, M. (2003). Learning from Different Teachers. *Machine Learning*, 51, 137-163.
- Arkin, R. C. (1999). *Behavior-Based Robotics* (2 ed.). Cambridge, MA: The MIT Press.
- Arkin, R. C., Fujita, M., Takagi, T., & Hasegawa, R. (2003). An ethological and emotional basis for human-robot interaction. *Robotics and Autonomous Systems*, 42, 191-201.
- Axelrod, R. (1984). *The Evolution of Cooperation*. New York: Basic Books.
- Bainbridge, W. S., Brent, E. E., Carley, K. M., Heise, D. R., Macy, M. W., Markovsky, B., et al. (1994). Artificial Social Intelligence. *Annual Review of Sociology*, 20, 407-436.
- Bandura, A. (1962). Social Learning Through Imitation. In M. R. Jones (Ed.), *Nebraska Symposium on Motivation* (pp. 211-269). Lincoln, NB: University of Nebraska.
- Banerjee, B., Mukherjee, R., & Sen, S. (2000). Learning mutual Trust. In *Proceedings of AGENTS-00 Workshop on Deception, Fraud and Trust in Agent Societies*, Utrecht, the Netherlands.
- Bar-On, R., Tranel, D., Denburg, N. L., & Bechara, A. (2003). Exploring the neurological substrate of emotional and social intelligence. *Brain*, 126(8), 1790-1800.
- Barber, B. (1983). *The Logic and Limits of Trust*. New Brunswick, New Jersey: Rutgers University Press.
- Bargh, J. A., Chen, M., & Burrows, L. (1996). Automaticity of social behavior: direct effects of trait construct and stereotype activation on action. *Journal of Personality and Social Psychology*, 71, 230-244.
- Bentivegna, D. C., Atkeson, C. G., & Cheng, G. (2004). Learning tasks from observation and practice. *Robotics and Autonomous Systems*, 47, 163-169.
- Berg, J., Dickhaut, J., & McCabe, K. (1995). Trust, Reciprocity, and Social History. *Games and Economic Behavior*, 10, 122-142.

- Berger, C. R. (1987). Communicating under Uncertainty. In M. E. Roloff & G. R. Miller (Eds.), *Interpersonal Processes: New Directions in Communication Research* (pp. 39-62). Newbury Park: Sage Publications.
- Bergman, T. J., Beehner, J. C., Cheney, D. L., & Seyfarth, R. M. (2003). Hierarchical Classification by Rank and Kinship in Baboons. *Science*, *302*(5648), 1234-1236.
- Beth, T., Borcharding, M., & Klein, B. (1994). Valuation of Trust in Open Networks. In *Proceedings of European Symposium on Research in Computer Security* Brighton, UK, pp. 3-18.
- Bethel, C. L., & Murphy, R. R. (2008). NEWHRI Workshop Abstract (ICRA '08) Directions of Focus in HRI. In *Proceedings of NEWHRI Workshop*, Pasadena, CA.
- Biernat, M., & Kobrynowicz, D. (1997). Gender- and race-based standards of competence: lower minimum standards but higher ability standards for devalued groups. *Journal of Personality and Social Psychology*, *72*, 544-557.
- Billard, A., Epars, Y., Calinon, S., Schaal, S., & Cheng, G. (2004). Discovering optimal imitation strategies. *Robotics and Autonomous Systems*, *47*, 69-77.
- Bodenhausen, G. V., Macrae, C. N., & Garst, J. (1998). Stereotypes in thought and deed: social-cognitive origins of intergroup discrimination. In C. Sedikides, J. Schopler & C. A. Insko (Eds.), *Intergroup Cognition and Intergroup Behavior* (pp. 311-336). Mahwah, NJ: Erlbaum.
- Bonasso, R. P., Kortenkamp, D., & Murphy, R. (1998). Mobile Robots: A Proving Ground for Artificial Intelligence. In D. Kortenkamp, R. P. Bonasso & R. Murphy (Eds.), *Artificial intelligence and mobile robots: Case Studies of successful robot systems* (pp. 3-18). Menlo Park, CA: American Association for Artificial Intelligence.
- Breazeal, C. L. (2002). *Designing Sociable Robots*. Cambridge, MA: The MIT Press.
- Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, *47*, 139-159.
- Byrne, R. W., & Whiten, A. (1997). Machiavellian intelligence. In A. Whiten & R. W. Byrne (Eds.), *Machiavellian Intelligence II: Extensions and Evaluations* (pp. 1-23). Cambridge: Cambridge University Press.
- Carpin, S., Wang, J., Lewis, M., Birk, A., & Jacoff, A. (2005). High Fidelity Tools for Rescue Robotics: Results and Perspectives. In *Proceedings of RoboCup 2005*, Osaka, Japan.

- Castelfranchi, C., & Falcone, R. (2000, January). Trust is much more than subjective probability: Mental components and sources of trust. In *Proceedings of Hawaii International Conference on System Sciences*, Kauai, Hawaii January.
- Castelfranchi, C., & Falcone, R. (2001). Social Trust: A Cognitive Approach. In C. Castelfranchi & Y.-H. Tan (Eds.), *Trust and Deception in Virtual Societies* (pp. 55-90).
- Chadwick-Jones, J. K. (1976). *Social Exchange Theory: Its structure and influence in social psychology*. London: Academic Press.
- Cohen, P. R., & Feigenbaum, E. (1982). *The handbook of artificial intelligence*. Reading, MA.
- Crandall, J. W., & Goodrich, M. A. (2004). Multiagent Learning During On-Going Human-Machine Interactions: The Role of Reputation. In *Proceedings of In AAAI Spring Symposium: Interaction between Humans and Autonomous Systems over Extended Operation*, Stanford, CA.
- Cross, R., & Borgatti, S. P. (2000). The Ties that Share: Relational Characteristics that Facilitate Information Seeking. In M. Huysman & V. W. (eds.) (Eds.), *Social Capital and Information Technology* (pp. 137-162). Cambridge MA MIT-Press.
- Damasio, A. R. (1994). *Descartes Error: Emotion, Reason, and the Human Brain*. New York, NY: Putnam and Sons.
- Davis, R., Shrobe, H., & Szolovits, P. (1993). What is a Knowledge Representation? *AI Magazine*, 14(1), 17-33.
- Deception. (1999). In *Webster's Dictionary*.
- Deutsch, M. (1962). Cooperation and Trust: Some Theoretical Notes. In M. R. Jones (Ed.), *Nebraska Symposium on Motivation* (pp. 275-315). Lincoln, NB: University of Nebraska.
- Deutsch, M. (1973). *The Resolution of Conflict: Constructive and Destructive Processes*. New Haven, CT: Yale University Press.
- Duck, S. (1973). *Personal Relationships and Personal Constructs: A study of friendship formation*. Oxford, England: John Wiley & Sons.
- Duck, S., Acitelli, L. K., Manke, B., & West, L. (2000). Sewing Relational Seeds: Contexts for Relating in Childhood. In R. S. L. Mills & S. Duck (Eds.), *The Developmental Psychology of Personal Relationships* (pp. 1-14). Chichester: John Wiley & Sons.
- Dyad. (2006). In *Random House Unabridged Dictionary*: Random House, Inc.

- Economist. (2006). Trust me, I'm a robot. *The Economist*, 379, 18-19.
- Falcone, R., & Castelfranchi, C. (2001). The socio-cognitive dynamics of trust: Does trust create trust? *Trust in Cyber-Societies: Integrating the Human and Artificial Perspectives*, pp.55-72.
- Farrington, D. P. (1993). Childhood origins of teenage antisocial behaviour and adult social dysfunction. *Journal of the Royal Society of Medicine*, 86(1), 13-17.
- Feil-Seifer, D. (2008). Socially Assistive Robot-Based Intervention for Children with Autism Spectrum Disorder. In *Proceedings of NEWHRI Workshop*, Pasadena, CA.
- Field, T. M., & Walden, T. A. (1982). Production and Discrimination. *Child Development*, 53, 1299-1300.
- Fong, T., Nourbakhsh, I., & Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42, 143-166.
- Gambetta, D. (1990). Can We Trust Trust? In D. Gambetta (Ed.), *Trust, Making and Breaking Cooperative Relationships* (pp. pages 213--237). Oxford England: Basil Blackwell.
- Gardner, H. (1983). *Frames of the Mind: The Theory of Multiple Intelligences*. New York, NY: Basic Books.
- Gibbons, R. (1992). *Game Theory for Applied Economists*. Princeton, NJ: Princeton University Press.
- Goffman, E. (1959). *The Presentation of Self in Everyday Life*. Garden City, NY: Doubleday.
- Good, D. A. (1991). Cooperation in a microcosm: lessons from laboratory games. In R. A. Hinde & J. Groebel (Eds.), *Cooperation and Prosocial Behavior* (pp. 224-237). Cambridge: Cambridge University Press.
- Green, M. (1986). A Survey of Three Dialogue Models. *ACM Transactions in Graphics*, 3(3), 244-275.
- Greenough, W. T., Black, J. E., & Wallace, C. S. (1987). Experience and Brain Development. *Child Development*, 58, 539-559.
- Griffin, E. (1997). *A First Look at Communication Theory* (Third Ed. ed.). New York, New York: McGraw-Hill Inc.
- Harnad, S. (1990). The Symbol Grounding Problem. *Physica D*, 42, 335-346.

- Heckel, F., & Smart, W. D. (2008). Mapping the Field of Human-Robot Interaction. In *Proceedings of NEWHRI Workshop*, Pasadena, CA.
- Henrich, J., Boyd, R., Bowles, S., Camerer, C., Fehr, E., & Gintis, H. (2003). *Foundations of Human Sociality: Ethnography and Experiments in Fifteen Small-scale Societies*. Oxford: Oxford University Press.
- Holmes, J. G., & Rempel, J. K. (1989). Trust in Close Relationships. In C. Hendrick (Ed.), *Close Relationships* (pp. 187-220). Newbury Park, CA: Sage Publications.
- Horswill, I. (1998). The Polly System: Case Studies of Successful Robot Systems. In D. Kortenkamp, R. P. Bonasso & R. Murphy (Eds.), *Artificial Intelligence and Mobile Robots* (pp. 122-139). Menlo Park, CA: American Association for Artificial Intelligence.
- Hsieh, M. A., Cowley, A., Keller, J. F., Chaimowicz, L., Grocholsky, B., Kumar, V., et al. (2007). Adaptive teams of Autonomous Aerial and Ground Robots for Situational Awareness. *Journal of Field Robotics*, 24(11-12), 991-1014.
- Humphrey, N. K. (1976). The social function of intellect. In P. P. G. Bateson & R. A. Hinde (Eds.), *Growing Points in Ethology* (pp. 303-317).
- Hung, Y. C., Dennis, A. R., & Robert, L. (2004). Trust in Virtual Teams: Towards an Integrative Model of Trust Formation. In *Proceedings of International Conference on System Sciences*, Hawaii.
- Hutchins, E. (1995). How a cockpit remembers its speeds. *Cognitive Science*, 19, 265-288.
- Isbell, C. L., Shelton, C., Kearns, M., Singh, S., & Stone, P. (2001). A Social Reinforcement Learning Agent. In *Proceedings of Agents*, Montreal, Canada.
- Jackson, J., & Tomkins, A. (1992). A Computational Model of Teaching. In *Proceedings of Proceedings of the fifth annual workshop on Computational Learning Theory*, Pittsburgh, PA.
- Josang, A. (2002). A Logic for Uncertain Probabilities *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3), 279-311.
- Josang, A., & Pope, S. (2005, January-February 2005). Semantic Constraints for Trust Transitivity. In *Proceedings of Second Asia-Pacific Conference on Conceptual Modeling* Newcastle, Australia January-February 2005.
- Josang, A., & Presti, S. L. (2004). *Analysing the Relationship between Risk and Trust*. Paper presented at the Second International Conference on Trust Management, Oxford.

- Karat, C.-M., Vergo, J., & Nahamoo, D. (2007). Conversational Interface Technologies. In A. Sears & J. A. Jacko (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*: Lawrence Erlbaum Associates Inc.
- Kelley, H. H. (1979). *Personal Relationships: Their Structures and Processes*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Kelley, H. H. (1984). The Theoretical Description of Interdependence By Means of Transition Lists. *Journal of Personality and Social Psychology*, 47(5), 956-982.
- Kelley, H. H., Holmes, J. G., Kerr, N. L., Reis, H. T., Rusbult, C. E., & Lange, P. A. M. V. (2003). *An Atlas of Interpersonal Situations*. New York, NY: Cambridge University Press.
- Kelley, H. H., & Thibaut, J. W. (1978). *Interpersonal Relations: A Theory of Interdependence*. New York, NY: John Wiley & Sons.
- King-Casas, B., Tomlin, D., Anen, C., Camerer, C. F., Quartz, S. R., & Montague, P. R. (2005). Getting to Know You: Reputation and Trust in Two-Person Economic Exchange. *Science*, 308, 78-83.
- Kitcher, P. (1982). *Abusing Science: The Case Against Creationism*. Cambridge: The MIT Press.
- Knight, J. (2001). Trust as a Form of Social Intelligence. In K. S. Cook (Ed.), *Trust in Society* (pp. 354-373). New York, NY: Russell Sage Foundation.
- Kollock, P. (1994). The Emergence of Exchange Structures: An Experimental Study of Uncertainty, Commitment, and Trust. *American Journal of Sociology*, 100(2), 313-345.
- Kortenkamp, D., Huber, M., Cohen, C., Raschke, U., Koss, F., & Congdon, C. (1998). Integrating High-Speed Obstacle Avoidance, Global Path Planning, and Vision Sensing on a Mobile Robot. In D. Kortenkamp, R. P. Bonasso & R. Murphey (Eds.), *Artificial Intelligence and Mobile Robots* (pp. 53-71). Menlo Park, CA: American Association for Artificial Intelligence.
- Kurban, R., & Houser, D. (2005). Experiments investigating cooperative types in humans: A complement to evolutionary theory and simulations. *Proceedings of the National Academy of Sciences*, 102(5), 1803-1807.
- Lacey, G., & Dawson-Howe, K. M. (1998). The application of robotics to a mobility aid for the elderly blind. *Robotics and Autonomous Systems*, 23, 245-252.
- Lee, J. D., & See, K. A. (2004). Trust in Automation: Designing for Appropriate Reliance. *Human Factors*, 46(1), 50-80.

- Lemmetty, S. (1999). *Review of Speech Synthesis*. Helsinki University of Technology, Helsinki.
- Lewicki, R. J., & Bunker, B. B. (1996). Developing and Maintaining Trust in Work Relationships. In R. M. Kramer & T. R. Tyler (Eds.), *Trust in Organizations* (pp. 114-139). Thousand Oaks, CA: SAGE Publications.
- Lewis, J. D., & Weigert, A. (1985). Trust as a Social Reality. *Social Forces*, 63(4), 967-985.
- Lin, L.-J. (1992). Self-Improving Reactive Agents Based On Reinforcement Learning, Planning, Teaching. *Machine Learning*, 8(3-4), 293-321.
- Luhmann, N. (1979). *Trust and Power*. Chichester: Wiley Publishers.
- Luhmann, N. (1990). Familiarity, Confidence, Trust: Problems and Alternatives. In D. G. (ed.) (Ed.), *Trust, Making and Breaking Cooperative Relationships*. Oxford, England: Basil Blackwell.
- Luna-Reyes, L., Cresswell, A. M., & Richardson, G. P. (2004). Knowledge and the Development of Interpersonal Trust: a Dynamic Model. In *Proceedings of International Conference on System Science*, Hawaii.
- Lund, M. (1991). Commitment old and new: social pressure and individual choice in making relationships last. In R. A. Hinde & J. Groebel (Eds.), *Cooperation and Prosocial Behavior* (pp. 212-223). Cambridge: Cambridge University Press.
- MacFarland, D., & Bossert, T. (1993). *Intelligent Behavior in Animals and Robots*. Cambridge, MA: The MIT Press.
- MacKenzie, D., Arkin, R., & Cameroon, J. (1997). Multiagent mission specification and execution. *Autonomous Robotics*, 4(1), 29-52.
- Macrae, C. N., & Bodenhausen, G. V. (2000). Social Cognition: Thinking Categorically about Others *Annual Review of Psychology*, 51, 93-120.
- Marsh, S. (1994). *Formalising Trust as a Computational Concept*. University of Stirling.
- Mataric, M. J., Eriksson, J., Feil-Seifer, D., & Winstein, C. (2007). Socially Assistive Robotics for Post-Stroke Rehabilitation. *International Journal of NeuroEngineering and Rehabilitation*, 4(5), 31-40.
- McCabe, K., Houser, D., Ryan, L., Smith, V., & Trouard, T. (2001). A functional imaging study of cooperation in two-person reciprocal exchange. *Proceedings of the National Academy of Sciences*, 98(20), 11832-11835.
- McCleskey, E. (1991). *Applying Deception to Special Operations Direct Action Missions*. Washington, D.C. Defense Intelligence College.

- Microsoft Speech SDK 5.1 information page. (2006). *Speech SDK 5.1*. Retrieved August 20, 2006 from <http://www.microsoft.com/speech/download/sdk51/>.
- Milgram, S. (1974). *Obedience to Authority: An Experimental View*. New York, NY: Harper and Row.
- Mitchell, T. (1997). *Machine Learning*. New York, NY: McGraw Hill.
- Norman, D. (1983). Some Observations on Mental Models. In D. Gentner & A. Stevens (Eds.), *Mental Models*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Nourbakhsh, I. (1998). Dervish: An Office-Navigating Robot. In D. Kortenkamp, R. P. Bonasso & R. Murphy (Eds.), *Artificial Intelligence and Mobile Robots* (pp. 73-90). Menlo Park, CA: American Association for Artificial Intelligence.
- Open Mind Speech. (2006). *Open Mind Speech*. Retrieved August 20, 2006 from <http://freespeech.sourceforge.net/>.
- Ortony, A., Clore, G. L., & Collins, A. (1988). *The Cognitive Structure of Emotions*. Cambridge, UK: Cambridge University Press.
- Osborne, M. J., & Rubinstein, A. (1994). *A Course in Game Theory*. Cambridge, MA: MIT Press.
- Perry, B. D. (2001). The neurodevelopmental impact of violence in childhood. In D. Schetky & E. Benedek (Eds.), *Textbook of child and adolescent forensic psychiatry* (pp. 221-238). Washington D.C.: American Psychiatric Press.
- Perry, B. D., & Pollard, D. (1997). Altered brain development following global neglect in early childhood. In *Proceedings of Society of Neuroscience: Proceedings from Annual Meeting*, New Orleans.
- Pettit, G. S., & Clawson, M. A. (1996). Pathways to Interpersonal Competence: Parenting and Children's Peer Relations. In N. Vanzetti & S. Duck (Eds.), *A Lifetime of Relationships* (pp. 126-153). Pacific Grove, CA: Brook/Cole Publishing
- Philipose, M., Fishkin, K. P., Perkowitz, M., Patterson, D. J., Fox, D., Kautz, H., et al. (2004). Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 50-57.
- Picard, R. (2000). *Affective Computing*. Cambridge, MA: The MIT Press.
- Pineau, J., Montemerlo, M., Pollack, M., Roy, N., & Thrun, S. (2003). Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems*, 42, 271-281.
- Popper, K. (1963). Conjectures and Refutations. In *Readings in the Philosophy of Science*. Mountain View, CA: Mayfield Publishing Company.

- Powers, A., & Kiesler, S. (2006). The advisor robot: tracing people's mental model from a robot's physical attributes. In *Proceedings of 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, Salt Lake City, UT, USA.
- Prietula, M. (2001). Advice, Trust, and Gossip Among Artificial Agents. In A. Lomi & E. Larson (Eds.), *Dynamics of Organizations: Computational Modeling and Organizational Theories*. Cambridge, MA: MIT Press.
- Prietula, M. J., & Carley, K. M. (2001). Boundedly Rational and Emotional Agents. In C. Castelfranchi & Y.-H. Tan (Eds.), *Trust and Deception in Virtual Society* (pp. 169-194): Kluwer Academic Publishers
- Quervain, D. J. F. d., Fischbacher, U., Treyer, V., Schellhammer, M., Schnyder, U., Buck, A., et al. (2004). The Neural Basis of Altruistic Punishment. *Science*, 305, 1254-1258.
- Quinlan, J. R. (1994). C4.5: Programs for Machine Learning. *Machine Learning*, 16(3), 235-240.
- Rabbie, J. M. (1991). Determinants of instrumental intra-group cooperation. In R. A. Hinde & J. Groebel (Eds.), *Cooperation and Prosocial Behavior* (pp. 238-262). Cambridge: Cambridge University Press.
- Relationship. (2000). In *American Heritage Dictionary*.
- Richards, Z., & Hewstone, M. (2001). Subtyping and Subgrouping: Processes for the Prevention and Promotion of Stereotype Change. *Personality and Social Psychology Review*, 5(1), 52-73.
- Rilling, J. K., Sanfey, A. G., Aronson, J. A., Nystrom, L. E., & Cohen, J. D. (2004). The neural correlates of theory of mind within interpersonal interactions. *NeuroImage*, 22, 1694-1703.
- Risk. (2007). In *Wikipedia, The Free Encyclopedia*. Retrieved 00:52, March 14, 2007, from <http://en.wikipedia.org/w/index.php?title=Risk&oldid=114591370>.
- Robert, A. J., Clark, K. R., & King, S. (2004). *Festival 2 - build your own general purpose unit selection speech synthesiser*. Paper presented at the International Speech Communication Association workshop on speech synthesis, Pittsburgh, PA.
- Rogers, E., & Murphy, R. (2001). Human-Robot Interaction Final Report for DARPA/NSF Study on Human-Robot Interaction
- Rousseau, D. M., Sitkin, S. B., Burt, R. S., & Camerer, C. (1998). Not so Different After All: A Cross-Discipline View of Trust. *Academy of Management Review*, 23(3), 393-404.

- Rusbult, C. E., & Van Lange, P. A. M. (2003). Interdependence, Interaction and Relationship. *Annual Review of Psychology*, *54*, 351-375.
- Russen, A. E. (1997). Exploiting the expertise of others. In A. Whiten & R. W. Byrne (Eds.), *Machiavellian Intelligence II: Extensions and Evaluations* (pp. 174-206). Cambridge: Cambridge University Press.
- Sabater, J., & Sierra, C. (2005). Review of Computational Trust and Reputation Models. *Artificial Intelligence Review*, *24*, 33-60.
- Salzinger, S., Feldman, R. S., & Hammer, M. (1993). The Effects of Physical Abuse on Children's Social Relationships. *Child Development*, *64*, 169-187.
- Sanfey, A. G. (2007). Social Decision-Making: Insights from Game Theory and Neuroscience. *Science*, *318*, 598-602.
- Sanfey, A. G., Rilling, J. K., Aronson, J. A., Nystrom, L. E., & Cohen, J. D. (2003). The Neural Basis of Economic Decision-Making in the Ultimatum Game. *Science*, *300*, 1755-1758.
- Scassellati, B. (2000). How Developmental Psychology and Robotics Complement Each Other. In *Proceedings of NSF/DARPA Workshop on Development and Learning*, Michigan State University, Lansing, MI, 2000.
- Scassellati, B. (2002). Theory of mind for a humanoid robot. *Autonomous Robots*, *12*, 13-24.
- Schaal, S. (1999). Is Imitation Learning the Route to Humanoid Robots. *Tend in Cognitive Sciences*, *3*, 233-242.
- Schatzman, L., & Strauss, A. (1955). Social Class and Modes of Communication. *The American Journal of Sociology*, *60*(4), 329-338.
- Schillo, M., & Funk, P. (1999). Learning from and About Other Agents in Terms of Social Metaphors. In *Proceedings of IJCAI Workshop on Agents Learning About, From and With other Agents*, Stockholm Sweden.
- Schillo, M., Funk, P., & Rovatsos, M. (2000). Using Trust for Detecting Deceitful Agents in Artificial Societies. *Applied Artificial Intelligence Journal, Special Issue on Trust, Deception and Fraud in Agent Societies*.
- Sears, D. O., Peplau, L. A., & Taylor, S. E. (1991). *Social Psychology*. Englewood Cliffs, New Jersey: Prentice Hall.
- Sharkey, N., & Sharkey, A. (2010 in press). The crying shame of robot nannies: an ethical appraisal. *Interaction Studies*.

- Shintaku, E., Fukui, K., Nishikawa, K., Takata, K., Takanishi, A., Takanobu, H., et al. (2005). Mechanical vocal cord model mimicking human biological structure. *The Journal of the Acoustical Society of America* 117(4), 2543.
- Shoval, S., Ulrich, I., & Borenstein, J. (2000). Computerized Obstacle Avoidance Systems for the Blind and Visually Impaired. In H. N. L. Teodorescu & L. C. Jain (Eds.), *Intelligent Systems and Technologies in Rehabilitation Engineering* (pp. 414-448): CRC Press.
- Situation. (2007). In *Encarta World English Dictionary, North American Edition*.
- Sony Corporation. (2006). ERS7-M3. In *Aibo Users Manual*. Retrieved August 20, 2006 from <http://www.sony.net/Products/aibo/>.
- Sternberg, R. J., Wagner, R. K., Williams, W. M., & Horvath, J. A. (1995). Testing Common Sense. *American Psychologist*, 50(11), 912-927.
- Stoytchev, A., & Arkin, R. C. (2001). Combining Deliberation, Reactivity, and Motivation in the Context of a Behavior-Based Robot Architecture. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA-2001)*, Banff, Canada.
- Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA: Bradford Books.
- Tesch, S. A., & Martin, R. R. (1983). Friendship concepts of Young Adults in Two Age Groups. *The Journal of Psychology*, 115, 7-12.
- Thrun, S., Bucken, A., Burgard, W., Fox, D., Frohlinghaus, T., Henning, D., et al. (1998). Map Learning and High-Speed Navigation in RHINO. In D. Kortenkamp, R. P. Bonasso & R. Murphy (Eds.), *Artificial Intelligence and Mobile Robots* (pp. 21-52). Menlo Park, CA: American Association for Artificial Intelligence.
- Thrun, S., Schulte, J., & Rosenberg, C. (2000). Interaction with Mobile Robots in Public Places. *IEEE Intelligent Systems*, 7-11.
- Toth, M., Halasz, J., Mikics, E., Barys, B., & Haller, J. (2008). Early social deprivation induces disturbed social communication and violent aggression in adulthood. *Behavioral neuroscience*, 122(4), 849-854.
- Travis, L., Sigman, M., & Ruskin, E. (2001). Links between Social Understanding and Social Behavior in Verbally Able Children with Autism. *Journal of Autism and Developmental Disorders*, 31(2), 119-130.
- Trigwell, K., Prosser, M., & Waterhouse, F. (1999). Relations between teachers' approaches to teaching and students' approaches to learning. *Higher Education*, 37, 57-70.

- Trivers, R. L. (1971). The evolution of reciprocal altruism. *Quarterly Review of Biology*, 46(189-225).
- Tversky, A., & Kahneman, D. (1992). Advances in prospect theory: Cumulative representations of uncertainty. *Journal of Risk and Uncertainty*, 5, 297-323.
- Velasquez, J., & Maes, P. (1997). Cathexis: A Computational Model of Emotions. In *Proceedings of Proceedings of the Fourteenth National Conference on Artificial Intelligence.*, Marina del Rey, CA, USA.
- Wagner, A. R. (2008, May). A Representation for Interaction. In *Proceedings of Proceedings of the ICRA 2008 Workshop: Social Interaction with Intelligent Indoor Robots (SI3R)*, Pasadena, CA, USA May.
- Wang, J., Lewis, M., Hughes, S., Koes, M., & Carpin, S. (2005). Validating usersim for use in hri research. In *Proceedings of Human Factors and Ergonomics Society 49th Annual Meeting (HFES'05)*, Orlando, FL.
- Werry, I., & Dautenhahn, K. (1999). Applying Mobile Robot Technology to the Rehabilitation of Autistic Children. In *Proceedings of Symposium on Intelligent Robotics Systems*, Coimbra, Portugal.
- Wright, D. E. (1999). *Personal Relationships: An Interdisciplinary Approach*. Mountain View, CA: Mayfield Publishing Company.
- Yamagishi, T. (2001). Trust as a Form of Social Intelligence. In K. S. Cook (Ed.), *Trust in Society* (pp. 121-147). New York, NY: Russell Sage Foundation.
- Zaratti, M., Fratarcangeli, M., & Iocchi, L. (2006). A 3D Simulator of Multiple Legged Robots based on USARSim. In *Proceedings of RoboCup Symposium 2006*, Bremen, Germany.
- Zender, H., Mozos, Ó. M., & Jensfelt, P. (2007). How to say "No" to a robot. Retrieved August 27, 2008, from www.dfki.de/~zender/