Patterns of Agent Behaviour

Multi-agent systems (Vrije Universiteit Amsterdam)

Part B Chapter 14

Integration Agent Models[1]

1 Introduction

The first type of integration of models addressed in this course is the integration of domain models and agent models. This integration takes place by embedding domain models in certain ways within agent models. By incorporating domain models within an agent model, the agent gets an understanding of the processes of its surrounding environment, which is a solid basis for knowledgeable intelligent behaviour. Four different ways to integrate domain models within agent models are considered in this course. A most simple way is to use a domain model that specifically models human behaviour in the following manner:

domain model directly used as agent model

In this case a domain model that describes human processes and behaviour is used directly as an agent model, in order to simulate human behaviour. Note that here the domain model and agent model refer to the same agent.

Such an agent model can be used in interaction with other agent models, in particular with *ambient agent models* to obtain a test environment for simulations. For this last type of (artificial) agents domain models can be integrated within their agent models in three different ways, in order to obtain one or more of the following (sub)models; see Figure 2.1. Here the solid arrows indicate information exchange between processes (data flow) and the dotted arrows the integration process of the domain models within the agent models.

analysis model

To perform analysis of the human's states and processes by reasoning based on observations (possibly using specific sensors) and the domain model.

support model

To generate support for the human by reasoning based on the domain model.

adaptation model

To tune parameters in the domain model better to the specific characteristics of the human by reasoning based on the domain model.

Note that here the domain model that is integrated refers to one agent (the human considered), whereas the agent model in which it is integrated refers to a different agent (the ambient software agent).

[1] By Zulfiqar A. Memon and Jan Treur, VU University Amsterdam, Behavioural Inforamatics Group.

Within this chapter each of the four types of models is briefly illustrated below by means of a simple imaginary example in which icecream plays a crucial role.

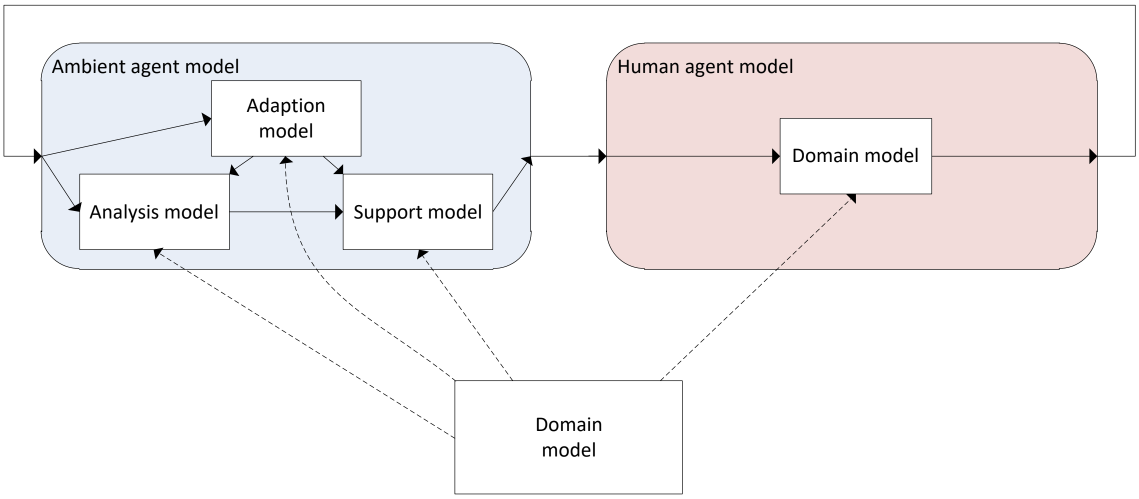

Figure 0.1 Four ways to integrate a domain model within an agent model

In Section 0 domain models are addressed. Section 0 discusses how domain models can be integrated within an agent model to obtain an analysis model. Section 0 does the same for support models. In Section 0 discusses integration of a domain model to obtain an adaptation model within an agent model.

2 Domain Models

Domain models considered here are models of parts of reality, including one or more humans and relevant aspect of their environment. Within the course Introduction to Modelling and Simulation many examples of domain models have been addressed, for example models for neural mechanisms for learning, adaptive decision making in human eating behaviour, motivation-based behaviour based on beliefs, desires and intentions, and trust-based intertemporal decision making as a basis for emergence of temporally reciprocal altruism. In the current course, each chapter first presents a domain model; examples are domain models for emotion generation, driving behaviour, emotion contagion within groups, criminal behaviour. Modelling techniques used for these domain models are often temporal logical approaches or dynamical systems approaches or combinations thereof. All of these techniques will be expressed in the hybrid LEADSTO format allowing both logical and numerical elements.

2.1 Analysis of the Main Aspects and their Relations for Example Domain Model

For an example (that will also be used in the subsequent sections), suppose that it is observed that *Arnie* crosses the street, and he is heading to a place *P* where they sell delicious icecream. The main concepts used to model this behaviour are:

A desires to buy icecream
A observes that at *P* they sell icecream

A goes to *P*
A is hungry
A does not eat
A likes icecream

The processes leading to such behaviour may involve a desire to buy icecream and a state of being hungry, as described in a simple domain model expressed by the following global dynamic relationships (see also Figure 2.2):

Affecting being hungry
A does not eat affects that A is hungry
Affecting the desire to buy icecream
A is hungry affects that A desires to buy icecream
A observes that at P they sell icecream affects that A desires to buy icecream
A likes icecream affects that A desires to buy icecream
Affecting the action
A desires to buy icecream affects that A goes to P
A observes that at P they sell icecream affects that A goes to P

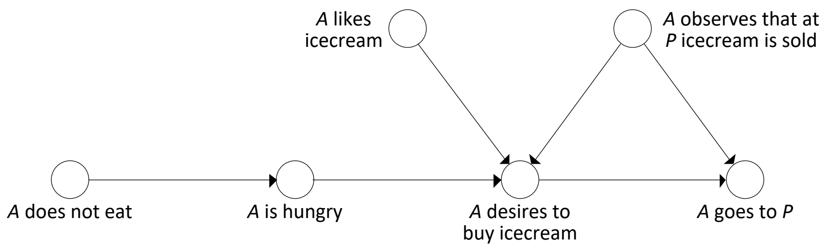

Figure 0.1 A simple example of a domain model

2.2 Detailed Domain Model for the Example

The concepts were formalised for the detailed domain model as shown in Table 0.1.

Concept	Formalisation
A desires to buy icecream	desire(A:AGENT, performed(A:AGENT, buy_icecream))
A observes that at *P* they sell icecream	observed(A:AGENT, icecream_at(P:POSITION))
A goes to *P*	performed(A:AGENT, goto(P:POSITION))
A is hungry	hungry(A:AGENT)
A likes icecream	likes(A:AGENT, icecream)
A does eat	performed(A:AGENT, eat)

Table 0.1 Formalisation of the concepts of the example domain model

The formalisation of some of the concepts makes use of *sorts*. These are presented in Table 0.2.

Sort	Description of use	Elements

AGENT	Ambient and human agents	a, b, a1, a2, ..,
ACTION	Actions	eat, buy_icecream, goto(P:POSITION)
FOOD	Types of food	icecream
POSITION	Positions	p1, p2, …
INFO_ELEMENT	Information element	icecream_at(P:POSITION)

Table 0.2 Sorts used

Within the language LEADSTO the dynamic relationships for the detailed domain model can be formalised as follows:

DDR1 Becoming hungry
If *A* does not eat
then *A* will be hungry
not performed(A, eat) $\twoheadrightarrow$ hungry(A)

DDR2 Generating the desire to buy icecream
If *A* is hungry
and *A* observes that at *P* they sell icecream
and *A* likes icecream
then *A* will desire to buy icecream
hungry(A) & observed(A, icecream_at(P)) & likes(A, icecream)
$\twoheadrightarrow$ desire(A, performed(A, buy_icecream))

DDR3 Generating the action to go to the icecream
If *A* desires to buy icecream
and *A* observes that at *P* they sell icecream
then *A* will go to *P*
desire(A, performed(A, buy_icecream)) & observed(A, icecream_at(P))
$\twoheadrightarrow$ performed(A, goto(P))

2.3 An Example Simulation Trace for the Domain Model

Based on the domain model described in Section 0 simulations can be performed. An example is shown in Figure 2.3. Here time is on the horizontal axis, and the formalised concepts (logical atoms indicating state properties) on the vertical axis. Dark lines indicate time intervals where the atoms are true and light lines where they are false. Otherwise they are unknown. What is shown in Figure 2.3 is that agent *a* always likes icecream, and that *a* did not eat from time point 0 on. As a result *a* is hungry from time point 1 on. After time point 1 *a* observes the icecream at *p1*, and (already being hungry) as a consequence after time point 2 generates the desire to buy icecream, which leads to the action to go to *p1* after time point 3.

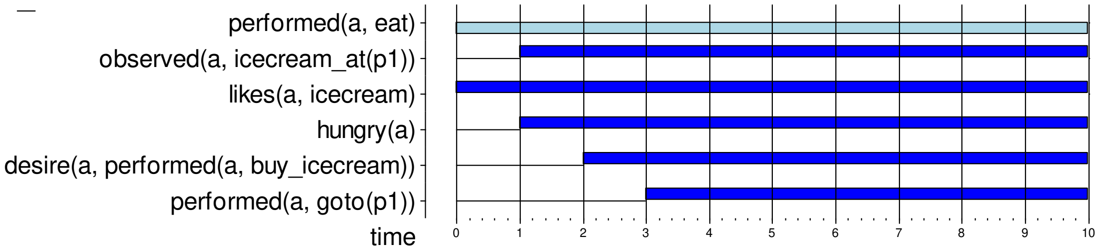

Figure 0.2 Example simulation trace for the domain model

3 Analysis models

A crucial element of ambient agents is that they collect and infer information about the human's functioning. Some aspects of the states and processes related to the human's functioning can be directly observed, but often many relevant aspects only can be indirectly derived. For such derivations it is useful to have a domain model integrated within the ambient agent model in the form of an *analysis model*.

3.1 Analysis of the Main Aspects and their Relations for an Example Analysis Model

For the example introduced above, suppose that it is observed that *Arnie* crosses the street, heading to a place where they sell delicious icecream, then one might derive the belief that *Arnie* desires to buy icecream. An analysis can also go further. For example, it can be found out why *Arnie* has this desire, and this may be related to beliefs that *Arnie* is hungry and that *Arnie* likes icecream, and by seeing the place to buy icecream this results in the desire to buy it. Moreover, one may derive that he is hungry because he did not eat. Such reasoning can be modelled when the domain model is integrated as an analysis model within the agent model for an ambient agent, *Xander* in the example. This integration can be done in different manners, in particular to enable *Xander* to perform reasoning forward in time or to perform reasoning backward in time. In the example, for the moment it is assumed that *Arnie* indeed likes icecream and *Xander* believes this; however, in Section 0 it is discussed how the model can be adapted when this is not a valid assumption. Moreover, it is assumed that *Xander* has as its goal to support *Arnie*, which entails (sub)goals such as the goal that *Arnie* should not become hungry. Therefore a belief that *Arnie* is becoming hungry can be used as a basis of the assessment that this internal state is undesirable.

CONCEPTS AND RELATIONS FOR REASONING FORWARD IN TIME IN AN ANALYSIS MODEL

The first option to use an analysis model is to use it for reasoning forward in time. For this the main concepts are:

ambient agent X observes Y
ambient agent X believes Y
ambient agent X believes that A does not eat
ambient agent X believes that A is hungry
ambient agent X believes that A desires to buy icecream
ambient agent X believes that A observes that at P they sell icecream
ambient agent X believes that A likes icecream

ambient agent X believes that A will go to P
ambient agent X desires that A is not hungry
ambient agent X assesses that it is undesirable that A is hungry

For reasoning forward in time the following format is used in the analysis model to represent integrated domain model relationships (see also Figure 2.4):

Affecting beliefs by observations
ambient agent X observes Y affects that ambient agent X believes Y
Affecting the belief that A is hungry
ambient agent X believes that A does not eat affects that ambient agent X believes that A is hungry
Affecting the belief that A desires to buy icecream
ambient agent X believes that A is hungry affects that ambient agent X believes that A desires to buy icecream
ambient agent X believes that A observes that at P they sell icecream affects that ambient agent X believes that A desires to buy icecream
ambient agent X believes that A likes icecream affects that ambient agent X believes that A desires to buy icecream
Affecting the belief that A goes to P
ambient agent X believes that A desires to buy icecream affects that ambient agent X believes that A will go to P
ambient agent X believes that A observes that at P they sell icecream affects that ambient agent X believes that A will go to P
Affecting the assessment that A is in an undesirable state of being hungry
ambient agent X believes that A is hungry affects that ambient agent X assesses that it is undesirable that A is hungry
ambient agent X desires that A is not hungry affects that ambient agent X assesses that it is undesirable that A is hungry

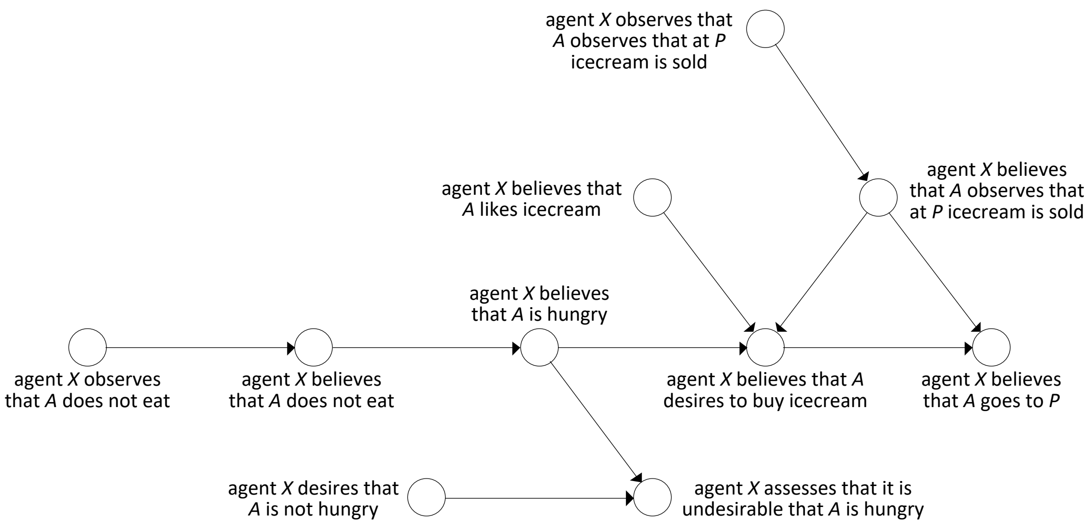

Figure 0.1 Example of an analysis model within an ambient agent for reasoning forward in time

Here *Y* can be instantiated by 'A observes a place P nearby where they sell delicious icecream', or 'A does not eat'. The other states of the human are assumed not observable. Note that the integration of the domain model relationships within such an analysis model for forward reasoning can be done in a systematic manner by embedding the concepts in a domain model relationship within beliefs or observations of the ambient agent; for example, the domain model relationship

E affects *F*

between concepts *E* and *F* (for example, *E* = '*A* is hungry', *F* = '*A* desires to buy icecream') is integrated within the analysis model as the relationship

ambient agent *X* believes that *E* affects ambient agent *X* believes that *F*

This analysis model can be used by the ambient agent to derive from the observation that *Arnie* does not eat via the belief that *Arnie* does not eat, the belief that *Arnie* will be hungry. From that belief, an assumed belief that *Arnie* likes icecream, and the belief that *Arnie* observes the place to buy icecream it can be derived that *Arnie* desires to buy icecream. From a belief that *Arnie* desires icecream and that *Arnie* observes icecream at *P*, it can be derived that *Arnie* will go to *P*.

The belief that *Arnie* is becoming hungry can be used as a basis of the assessment that this internal state is undesirable, if a belief is assumed that the ambient agent *Xander*'s desire is to let *Arnie* not become hungry.

CONCEPTS AND RELATIONS FOR REASONING BACKWARD IN TIME IN AN ANALYSIS MODEL

Also alternative options to integrate a domain model within the ambient agent model can be used. For the icecream example reasoning backward in time (a form of abduction) can be applied when the domain model relationships are integrated within the analysis model in the following (reverse) manner (see also Figure 2.5):

Affecting beliefs by observations

ambient agent X observes Y affects that ambient agent X believes Y

Affecting that ambient agent X believes that A desires to buy icecream

ambient agent X believes that A likes icecream affects that ambient agent X believes that A is hungry

ambient agent X believes that A goes to P affects that ambient agent X believes that A desires to buy icecream

ambient agent X believes that A observes that at P they sell icecream affects that ambient agent X believes that A desires to buy icecream

Affecting that ambient agent X believes that A is hungry

ambient agent X believes that A desires to buy icecream affects that ambient agent X believes that A is hungry

ambient agent X believes that A observes that at P they sell icecream affects that ambient agent X believes that A is hungry

Affecting the assessment that A is in an undesirable state of being hungry

ambient agent X believes that A is hungry affects that ambient agent X assesses that it is undesirable that A is hungry ambient agent X desires that A is not hungry affects that ambient agent X assesses that it is undesirable that A is hungry

Note that the integration of the domain model relationships within such an analysis model for backward reasoning can be done in a systematic manner by embedding the concepts in a domain model relationship within beliefs of the ambient agent in a reverse manner, for example, the domain relationship

E affects F

between concepts E and F (for example, E = ‘A desires to buy icecream’, F = ‘A goes to P’) is integrated within the analysis model as the relationship between beliefs about them, and reversing the direction for two of the concepts, thus obtaining:

ambient agent X believes that F affects ambient agent X believes that E

Such types of reasoning backward in time are what humans often perform when they try to figure out why others do what they see them doing, where this ‘why’ question is answered in terms of the human’s internal states that lead to the actions considered. By integrating a domain model as an analysis model within an ambient agent model in the way indicated, the ambient agent can also perform such reasoning backward in time. For the icecream example it can conclude the belief that Arnie is hungry from the belief that the action to go to P occurs. As before, this belief can be used as a basis of the assessment that this internal state of being hungry is undesirable, if a desire within the ambient agent Xander is assumed that Arnie is not hungry.

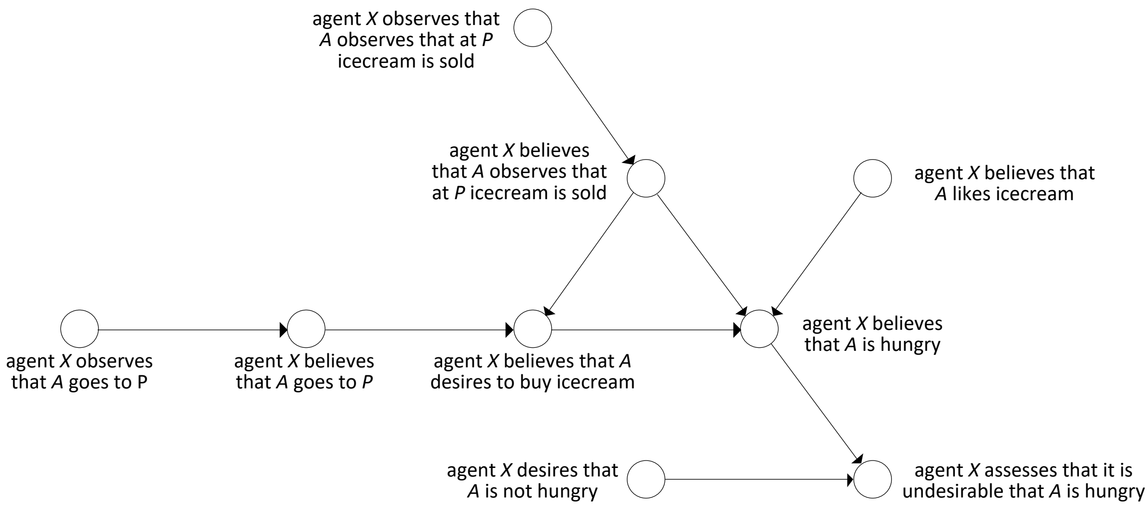

Figure 0.2 Example analysis model for reasoning backward in time

3.2 Detailed Analysis Model for the Example

For the detailed analysis models for the icecream example the main concepts have been formalised as shown in Table 0.1.

Concept	Formalisation
ambient agent X observes Y	observed(X: AGENT, Y:INFO_ELT)

ambient agent *X* believes that *A* desires to buy icecream	belief(X: AGENT, desire(A: AGENT, performed(A, buy_icecream)))
ambient agent *X* believes that *A* observed that at *P* they sell icecream	belief(X: AGENT, observed(A: AGENT, icecream_at(P:POSITION)))
ambient agent *X* believes that *A* goes to *P*	belief(X, performed(A: AGENT, goto(P:POSITION)))
ambient agent *X* believes that *A* is hungry	belief(X: AGENT, hungry(A: AGENT))
ambient agent *X* believes that *A* likes icecream	belief(X: AGENT, likes(A: AGENT, icecream))
ambient agent *X* believes that *A* does not eat	belief(X: AGENT, not(performed(A: AGENT, eat)))
ambient agent *X* desires that *A* is not hungry	desire(X: AGENT, not(hungry(A:AGENT)))
ambient agent *X* assesses that the undesirable state occurs that *A* is hungry	assessment(X: AGENT, undesirable_state(hungry(A:AGENT)))

Table 0.1 Formalisation of the concepts of the example analysis models

DETAILED ANALYSIS MODEL FOR REASONING FORWARD IN TIME

The example analysis model reasoning forward in time the domain model is integrated in the form of the following dynamic relationships:

ADR1 Generating beliefs from observations
If ambient agent X observes Y
then ambient agent X will believe Y
observed(X, Y) $\twoheadrightarrow$ belief(X, Y)

ADR2 Generating the belief that A is hungry
If ambient agent X believes that A does not eat
then ambient agent X will believe that A is hungry
belief(X, not(performed(A, eat))) $\twoheadrightarrow$ belief(X, hungry(A))

ADR3 Generating the belief that A desires to buy icecream
If ambient agent X believes that A is hungry
and ambient agent X believes that A observes that at P they sell icecream
and ambient agent X believes that A likes icecream
then ambient agent X will believe that A desires to buy icecream
belief(X, hungry(A)) &
belief(X, observes(A, icecream_at(P))) &
belief(X, likes(A, icecream))
$\twoheadrightarrow$ belief(X, desire(A, performed(A, buy_icecream)))

ADR4 Generating the belief that A goes to the icecream
If ambient agent X believes that A desires to buy icecream
and ambient agent X believes that A observes that at P they sell icecream
then ambient agent X will believe that A will go to P
belief(X, desire(A, performed(A, buy_icecream))) &
belief(X, observes(A, icecream_at(P)))
$\twoheadrightarrow$ belief(X, performed(A, goto(P)))

ADR5 Generating the assessment that A is in an undesirable state of being hungry
If ambient agent X believes that A is hungry
and ambient agent X desires that A is not hungry
then ambient agent X will assess that it is undesirable that A is hungry
belief(X, hungry(A)) & desire(X, not(hungry(A)))
↠ assessment(X, undesirable_state(hungry(A)))

The integration of the domain model relationships within such a formalised analysis model for reasoning forward in time is done in a systematic manner by replacing the atoms in a domain model relationship, for example

a & b ↠ c

by beliefs of the ambient agent about them, thus obtaining:

belief(X, a) & belief(X, b) ↠ belief(X, c)

DETAILED ANALYSIS MODEL FOR REASONING BACKWARD IN TIME

For the icecream example analysis model for reasoning backward in time (a form of abduction), the domain model is integrated in the form of the following dynamic relationships:

ADR6 Abducing the belief that A desires to buy icecream
If ambient agent X believes that A goes to P
and ambient agent X believes that A observes that at P they sell icecream
and ambient agent X believes that A likes icecream
then ambient agent X will believe that A desires to buy icecream
belief(X, performed(goto(P))) &
belief(X observed(A, icecream_at(P)))
belief(X likes(A, icecream))
↠ belief(X, desire(A, performed(A, buy_icecream)))

ADR7 Abducing the belief that A is hungry
If ambient agent X believes that A desires to buy icecream
and ambient agent X believes that A observes that at P they sell icecream
then ambient agent X will believe that A is hungry
belief(X, desire(A, performed(A, buy_icecream))) &
belief(X observed(A, icecream_at(P))) &
↠ belief(X, hungry(A))

The assessment is made by using dynamic relationship ADR5 from the detailed analysis model above.

The integration of the domain model relationships within a formalised analysis model for reasoning backward in time was done in a systematic manner by replacing the atoms in a domain model relationship, for example

a & b ↠ c

by beliefs about them, and reversing the direction for two of the atoms, thus obtaining:

belief(X, a) & belief(X, c) ↠ belief(X, b)

3.3 Example Traces for the Two Analysis Models

In Figure 2.6 an example trace for the analysis based on forward reasoning as performed by ambient agent *x* is shown (in addition to the trace for agent *a* based on the domain model shown in Figure 2.3). Right from the start, ambient agent *x* believes that *a* likes icecream, and it has a desire that *a* is not hungry. Next it observes that *a* does not eat and that *a* observes the icecream. From this it derives by reasoning forward in time that *a* is hungry and will have the desire to buy icecream. Finally, it assesses the state of *a* being hungry as undesirable, and that *a* will go to *p1*.

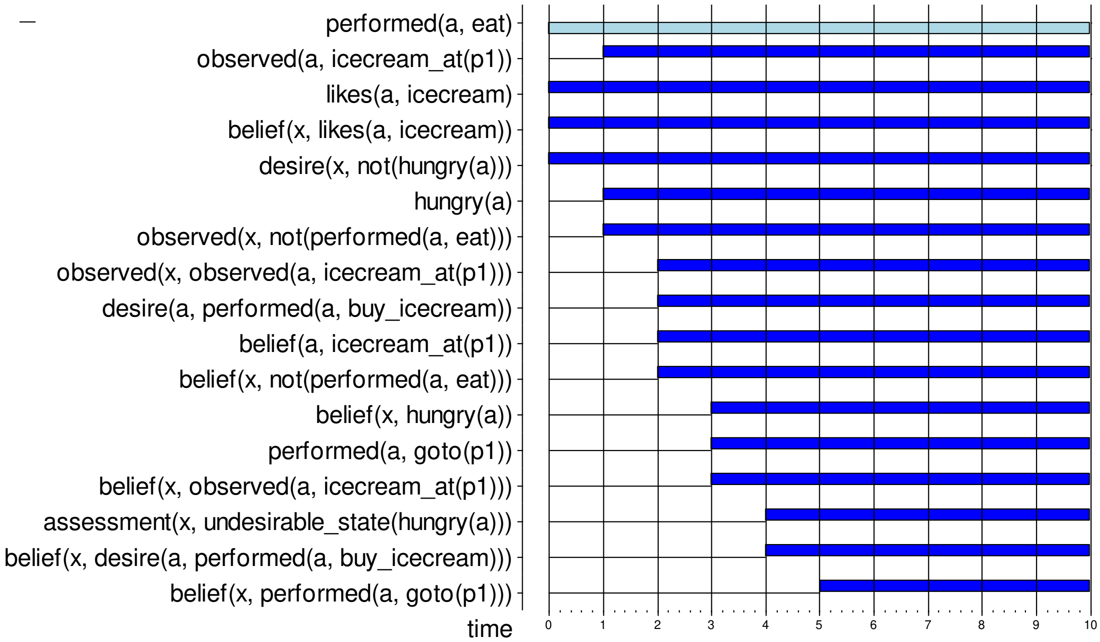

Figure 0.3 Example of an analysis model within an ambient agent for reasoning forward in time

In Figure 2.7 an example trace for the analysis based on backward reasoning as performed by ambient agent *x* is shown. Again right from the start, ambient agent *x* believes that *a* likes icecream, and it has a desire that *a* is not hungry. But it also has the beliefs (based on observations) that *a* is going to *p1* and that *a* has observed the icecream there. From this it derives by reasoning backward in time the belief that *a* has the desire to buy icecream and from this the belief that *a* is hungry. Finally, as for the forward reasoning case it assesses the state of *a* being hungry as undesirable.

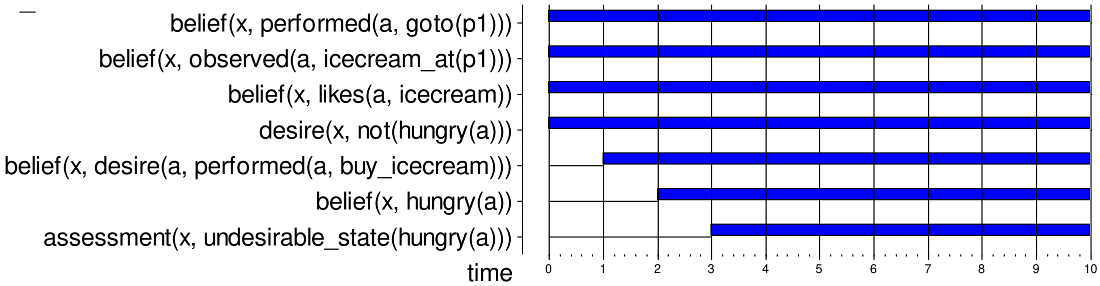

Figure 0.4 Example of an analysis model within an ambient agent for reasoning forward in time

4 Support Models

For an ambient agent to have some beliefs and assessments about the human's internal state is one thing, but to be of any help also actions are needed to avoid undesirable states. To generate actions that fit to the results of the analysis, a *support model* can be used by the ambient agent.

4.1 Analysis of the Main Concepts and their Relations for an Example Support Model

Suppose in the icecream example, it has been assessed as undesirable that Arnie is hungry. Moreover suppose that the ambient agents believes that there is a restaurant just around the corner. Then as an action the ambient agent may consider to propose Arnie to go to that restaurant for a healthy dinner. To be able to derive such a proposal, an extension of the domain model can be used, namely by (see Figure 2.8):

DDR4 From going to a restaurant to eating there
If A goes to a place Q
and at Q there is a restaurant
then A will eat a meal

performed(A, goto(Q)) & restaurant_at(Q) ↠ performed(A, eat)

DDR5 Not becoming hungry by eating
If A does eat
then A will not be hungry

performed(A, eat) ↠ not hungry(A)

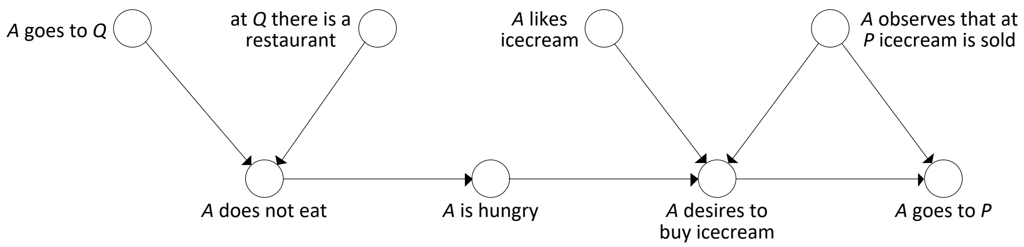

Figure 0.1 Example domain model extended

A SUPPORT MODEL FOR REASONING FORWARD IN TIME

The first approach for a support model is to generate action options and evaluate them one by one by assuming them and deriving their consequences by reasoning forward in time. In this way for the option for *A* to go to *Q* it can be concluded by the ambient agent that the action for *A* to go to *Q* will resolve the undesirable state for *A* of being hungry. To this end within the ambient agent model the extended domain model can be integrated as a support model as follows (see Figure 2.9). The main concepts considered are:

ambient agent X assumes that A goes to Q
ambient agent X believes that A goes to Q
ambient agent X believes that A eats a meal
ambient agent X believes that at Q there is a restaurant
ambient agent X believes that A is not hungry
ambient agent X proposes that A goes to Q
ambient agent X tells that there is a restaurant at Q

Global relationships between these concepts are (see Figure 2.9):

Affecting incorporating an assumption as belief
ambient agent X assumes that A goes to Q affects that ambient agent X believes that A goes to Q
Affecting the belief that A will eat in the restaurant
ambient agent X believes that A goes to at place Q affects that ambient agent X believes that A will eat
ambient agent X believes that at Q there is a restaurant affects that ambient agent X believes that A will eat
Affecting the belief that A will not be hungry
ambient agent X believes that A does eat a meal affects that ambient agent X believes that A will not be hungry
Affecting proposing A to go to Q
ambient agent X assumes that A goes to Q affects that ambient agent X proposes that A goes to Q
ambient agent X believes that A is not hungry affects that ambient agent X proposes that A goes to Q
ambient agent X believes that at Q there is a restaurant affects that ambient agent X proposes that A goes to Q
Affecting telling A about the restaurant at Q
ambient agent X assumes that A goes to Q affects that ambient agent X tells that there is a restaurant at Q
ambient agent X believes that A is not hungry affects that ambient agent X tells that there is a restaurant at Q
ambient agent X believes that at Q there is a restaurant affects that ambient agent X tells that there is a restaurant at Q

The integration of the domain model relationships within such a support model for forward reasoning can be done in a systematic manner by embedding the concepts in a domain model relationship within beliefs or observations of the ambient agent; for example, the domain model relationship

E affects *F*

between concepts *E* and *F* (for example, *E* = '*A* goes to *Q*', *F* = '*A* does eat') is integrated within the support model as the relationship

ambient agent *X* believes that *E* affects ambient agent *X* believes that *F*

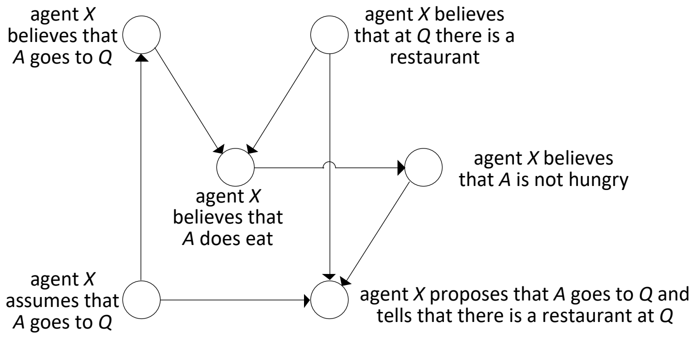

Figure 0.2 An example of a support model within an ambient agent based on reasoning forward in time

MODEL FOR REASONING BACKWARD IN TIME

The domain model can also be integrated as a support model for reasoning backward in time in the following way (see Figure 2.10). Here, the main idea is to take a desired state as a basis, and to derive other desired states from this, until a state is derived that can directly be influenced by the agent. The main concepts are:

ambient agent *X* desires that *A* is not be hungry
ambient agent *X* desires that *A* does eat
ambient agent *X* desires that *A* goes to *Q*
ambient agent *X* believes that there is a restaurant at *Q*
ambient agent *X* proposes that *A* goes to *Q*
ambient agent *X* tells that there is a restaurant at *Q*

The global relationships between the concepts are (see Figure 2.10):

Affecting the desire that A eats
ambient agent X desires that A is not be hungry affects that ambient agent X desires that A does eat
Affecting the desire that A goes to Q
ambient agent X desires that A does eat affects that ambient agent X desires that A goes to Q
ambient agent X believes that at Q there is a restaurant affects that ambient agent X desires that A goes to Q
Affecting proposing that A goes to Q
ambient agent X desires that A goes to Q affects that ambient agent X proposes that A goes to Q
ambient agent X believes that there is a restaurant at Q affects that ambient agent X proposes that A goes to Q
Affecting telling A about the restaurant at Q
ambient agent X desires that A goes to Q affects that ambient agent X tells that there is a restaurant at Q
ambient agent X believes that there is a restaurant at Q affects that ambient agent X tells that there is a restaurant at Q

The integration of the domain model relationships within such a support model for backward reasoning can be done in a systematic manner by embedding the concepts in a domain model relationship within desires of the ambient agent in a reverse manner, for example, the domain relationship

E affects *F*

between concepts *E* and *F* (for example, *E* = '*A* goes to *Q*', *F* = '*A* eats') is integrated within the support model as the relationship between desires about them, and reversing the direction for the atoms, thus obtaining:

ambient agent *X* desires that *F* affects ambient agent *X* desires that *E*

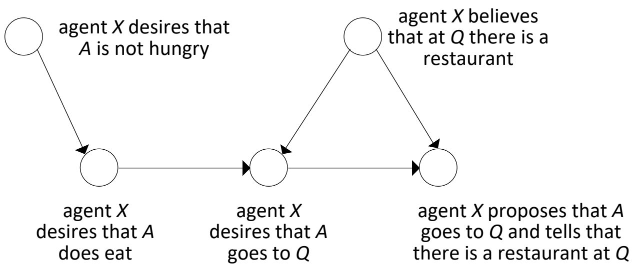

Figure 0.3 An example of a support model within an ambient agent for reasoning backward in time

4.2 Detailed Support Models for the Example

In this section detailed support models for the simple icecream example are introduced. In Table 0.1 it is shown how the main concepts were formalised.

Concept	Formalisation
ambient agent *X* believes that *A* goes to *Q*	belief(X: AGENT, performed(A: AGENT, goto(Q:POSITION)))
ambient agent *X* assumes that *A* goes to *Q*	assumption(X: AGENT, performed(A: AGENT, goto(Q:POSITION)))
ambient agent *X* believes that there is a restaurant at *Q*	belief(X: AGENT, restaurant_at(Q:POSITION))
ambient agent *X* believes that *A* eats	belief(X: AGENT, performed(A: AGENT, eat))
ambient agent *X* believes that *A* is not hungry	belief(X: AGENT, not(hungry(A: AGENT)))
ambient agent *X* proposes *A* that *A* goes to Q	proposed(X: AGENT, A:AGENT,
ambient agent *X* tells *A* that there is a restaurant at *Q*	performed(A: AGENT, goto(Q:POSITION)))
ambient agent *X* desires that *A* is not hungry	told(X: AGENT, A:AGENT, restaurant_at(Q:POSITION))
ambient agent *X* desires that *A* eats	desire(X: AGENT, not(hungry(A:AGENT)))

Table 0.1 Formalisation of the concepts of the example support models

A DETAILED SUPPORT MODEL FOR REASONING FORWARD IN TIME

The following specifies the detailed support model using reasoning forward in time.

SDR1 Incorporating an assumption as a belief

If ambient agent *X* assumes that *A* goes to *Q*
then ambient agent *X* will believe that *A* goes to *Q*

assumed(X, performed(A, goto(Q)))
$\twoheadrightarrow$ belief(X, performed(A, goto(Q)))

SDR2 Generating the belief that A will eat when at the restaurant

If ambient agent *X* believes that *A* goes to at place Q
and ambient agent *X* believes that at Q there is a restaurant
then ambient agent *X* will believe that *A* will eat a meal

belief(X, performed(A, goto(Q))) & belief(X, restaurant_at(Q))
$\twoheadrightarrow$ belief(X, performed(A, eat))

SDR3 Belief on not becoming hungry by eating

If ambient agent *X* believes that *A* does eat a meal
then ambient agent *X* will believe that *A* will not be hungry

belief(X, performed(A, eat))
$\twoheadrightarrow$ belief(X, not(hungry(A)))

SDR4 Proposing and telling A to go to the restaurant

If ambient agent *X* assumes that *A* goes to *Q*
and ambient agent *X* believes that *A* is not hungry
and ambient agent *X* believes that at Q there is a restaurant
then ambient agent *X* will propose that *A* goes to *Q*
and ambient agent *X* will tell that there is a restaurant at *Q*

assumed(X, performed(A, goto(Q))) & belief(X, not(hungry(A))) &
belief(X, restaurant_at(Q))
↠ proposed(X, A, performed(A, goto(Q))) & told(X, A, restaurant_at(Q))

The integration of the domain model relationships within such a support model for forward reasoning can be done in a systematic manner by replacing the atoms in a domain model relationship, for example

a & b ↠ c

by beliefs about them, thus obtaining:

belief(X, a) & belief(X, b) ↠ belief(X, c)

A DETAILED SUPPORT MODEL FOR REASONING BACKWARD IN TIME

The detailed support model using reasoning backward in time is specified by the following dynamic relationships.

SDR5 Generating the desire that *A* eats

If ambient agent *X* desires that *A* is not be hungry
then ambient agent *X* will desire that *A* does eat

desire(X, not(hungry(A)))
↠ desire(X, performed(A, eat))

SDR6 Generating the desire that *A* goes to *Q*

If ambient agent *X* desires that *A* does eat
and ambient agent *X* believes that at *Q* there is a restaurant
then ambient agent *X* will desire that *A* goes to *Q*

desire(X, performed(A, eat)) & belief(X, restaurant_at(Q))
↠ desire (X, performed(A, goto(Q)))

Based on the desire that *Arnie* goes to a restaurant for a dinner, the action can be generated, based on the following part of the support model (see Figure 2.10):

SDR7 Proposing and telling *A* to go to the restaurant

If ambient agent *X* desires that *A* goes to *Q*
and ambient agent *X* believes that there is a restaurant at *Q*
then ambient agent *X* will propose that *A* goes to *Q*
and ambient agent *X* will tell that there is a restaurant at *Q*

desire(X, performed(A, goto(Q))) & belief(X, restaurant_at(Q))
↠ proposed(X, A, performed(A, goto(Q))) & told(X, A, restaurant_at(Q))

Note that the integration of the domain model relationships within such a support model for backward reasoning can be done in a systematic manner by embedding most of the atoms in a desire (or belief) in a domain model relationship and reversing the order, for example, for a & b ↠ c this may obtain belief(X, a) & desire(X,c) ↠ desire(X,b)

4.3 Example Traces for the Two Support Models

In this section traces are shown for the two support models described in Section 0. First, in Figure 2.11 a trace is shown for the forward reasoning support model. Here as a starting point the assumption is made that *a* goes to *p2*, where the restaurant is. To evaluate this assumption, by forward reasoning the ambient agent derives as beliefs its consequences: the beliefs that *a* will eat and will not be hungry. Finally from that the proposal for *a* to go to *p2* is derived, and the information is given that there is a restaurant at *p2*.

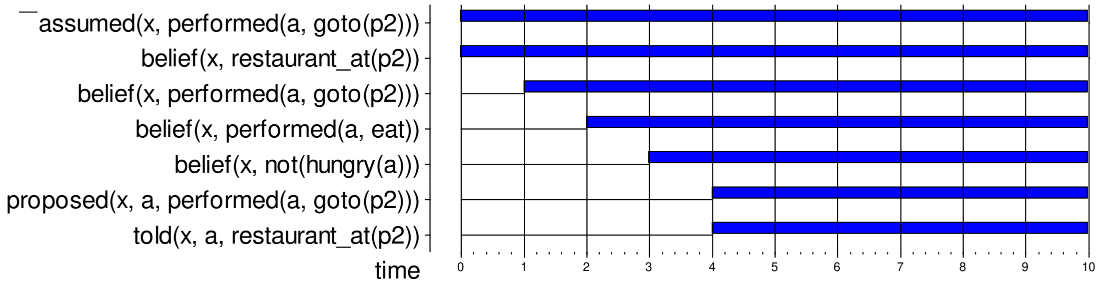

Figure 0.4 An example trace for the support model reasoning forward in time

In Figure 2.12 a trace is shown for the backward reasoning support model. Here from the desire that a is not hungry it is derived by backward reasoning the desire that *a* should eat. From this by backward reasoning it derives the desire that *a* will go to *p2*. Finally from that desire the proposal for *a* to go to *p2* is derived, and it is told that there is a restaurant at *p2*.

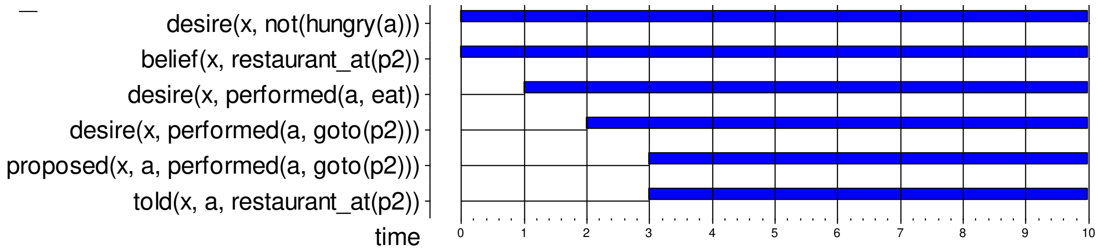

Figure 0.5 An example trace for the support model for reasoning backward in time

5 An adaptation Model

Domain models often include a number of *parameters* that represent certain characteristics of the processes modelled. In principle, for a given domain process such characteristics are assumed constant; they allow to tune a domain model to specific situations, for example, specific types of personalities of humans. An ambient agent may have beliefs about these characteristics and use them in its reasoning processes. An example is the characteristic of *Arnie* that he likes icecream; this was formalised as a parameter in a binary, logical manner. Characteristics may also be formalised as parameters in numerical manners, for example, a number indicating the extent to which *Arnie* likes icecream, or the growth factor of a population. For a domain model the parameters involved are assumed fixed, but beliefs about them may change over time.

Beliefs on parameters used by an ambient agent in its analysis and support models need not (exactly) correspond to the actual parameters for the domain model that correspond to reality. Within an ambient agent the beliefs about such parameters may be questioned in the light of observations made. For example, if a certain growth factor for a population is believed, and it is observed that after some time the population is much smaller than predicted on the basis of the believed growth factor, the ambient agent may want to adapt its belief on the growth factor by replacing this belief by a belief in a lower value. When beliefs on parameters used by the ambient agent substantially differ from reality, the agent may make errors in its analysis and in its support actions. Often it is not easy to estimate such parameters at forehand (at design time). Therefore it is better to design an ambient agent in such a way that it can learn by identifying errors in parameter beliefs, and after discovering them, adapting these beliefs in order to obtain more correct beliefs on the parameters (at runtime).

This section presents models for such an adaptation process within an ambient agent. Two main questions to be addressed are:

How does the agent get information on the (extent of) *deviation* of the model from reality?
How can it relate such identified deviations to adaptations required in parameter beliefs to compensate for them?

For an answer to the first question, in this section it is assumed that from time to time information becomes available, as *observations* for the ambient agent. A deviation can be obtained as a difference between observation and prediction on the basis of a prediction model similar to the analysis model using the given beliefs on the parameters.

For the second question a deviation found has to be related to the beliefs on the parameters. Although it often may be assumed that in the numerical case the larger the deviation, the more the parameter value has to be adapted, still it is not clear in which direction (positive or negative) and to which extent such an adaptation of a parameter value is needed. In particular, when more than one parameter is involved this may turn out a nontrivial challenge.

5.1 Main Aspects and their Relations for an Adaptation Model

A rough sketch for the qualitative case is as follows. As a first step *predictions* are made for a chosen *assumption* on a parameter. Such predictions are compared to *observations*. If they are different, then a *deviation* is found for the prediction. In this case the parameter assumption is rejected and the parameter belief is *adapted* to a different parameter belief.

Thus the following general concepts are used for the parameter adaptation model:

belief on parameters
assumptions on parameters
predictions for given parameter assumptions
observations
belief on deviations between observation and prediction

The following dynamic relationships between these concepts can be established (see also Figure 2.13 and Figure 2.14):

Affecting beliefs from observations (Figure 2.13) the ambient agent's observation of Y affects the ambient agent's belief on Y

Affecting predictions for parameter assumptions (Figure 2.13)

the ambient agent's beliefs affect the ambient agent's predictions

the ambient agent's parameter assumptions affect the ambient agent's parameter predictions

Affecting predictions following the integrated domain model relationships (Figure 2.13)

the ambient agent's prediction that A is hungry affects the ambient agent's prediction that A desires to buy icecream

the ambient agent's prediction that A likes icecream affects the ambient agent's prediction that A desires to buy icecream

the ambient agent's belief that A observed icecream at P affects the ambient agent's prediction that A desires to buy icecream

the ambient agent's prediction that A desires to buy icecream affects the ambient agent's prediction that A goes to P

the ambient agent's belief that A observed the icecream at P affects the ambient agent's prediction that A goes to P

Affecting deviation beliefs (Figure 2.14)

the ambient agent's observations affect the ambient agent's beliefs on deviations between observations and predictions

the ambient agent's predictions affect the ambient agent's beliefs on deviations between observations and predictions

Affecting adapted parameter beliefs (Figure 2.14)

the ambient agent's beliefs on deviations between observations and predictions affect the ambient agent's parameter beliefs

the ambient agent's assumptions on parameters affect the ambient agent's parameter beliefs

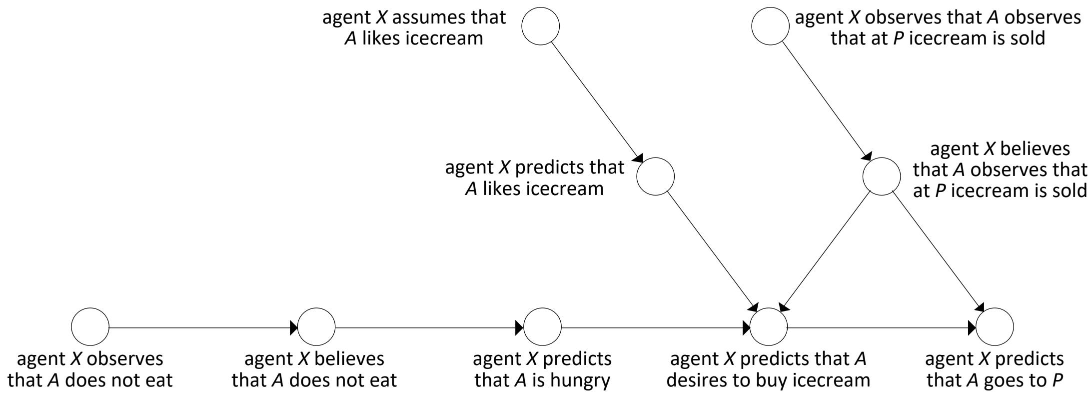

Figure 0.1 Example of a prediction model based on parameter assumptions for the icecream case

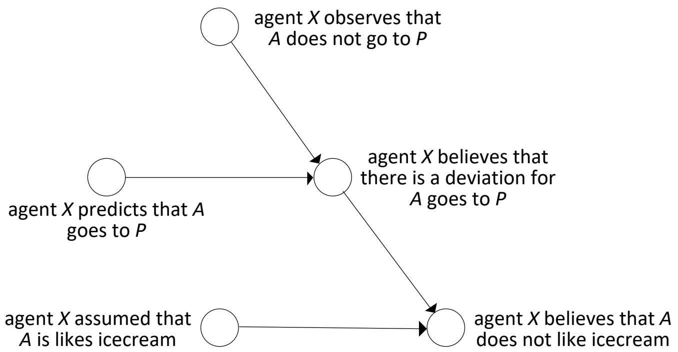

Figure 0.2 Example of a parameter belief adaptation model for the icecream case

The integration of the domain model relationships within a prediction model used for forward reasoning within the adaptation model can be done in a systematic manner by embedding the concepts in a domain model relationship within prediction statements, for example, from

E affects *F*

it is obtained:

ambient agent *X* predicts *E* affects ambient agent *X* predicts *F*

5.2 A Detailed Adaptation Model

In Table 0.1 a formalisation of the concepts can be found.

Concept	Formalisation
beliefs on parameter of liking icecream	belief(X:AGENT, likes(A:AGENT, icecream))
assumption on parameter of liking icecream	belief(X:AGENT, not(likes(A:AGENT, icecream)))
agent *X* observes *Y*	assumption(X:AGENT, likes(A:AGENT, icecream))
belief on not eating	observed(X:AGENT, Y:INFO_ELT)
belief on A's observation that icecream is sold	belief(X:AGENT, not(performed(A:AGENT, eat)))
prediction on liking icecream	belief(X:AGENT, observed(A:AGENT, icecream_at(P:POSITION)))
prediction on being hungry	prediction(X:AGENT, likes(A:AGENT, icecream))
prediction on desire to buy icecream	prediction(X:AGENT, hungry(A:AGENT))
prediction on going to *P*	prediction(X:AGENT, desire(A:AGENT,
belief on deviation between observation and prediction on going to *P*	performed(A:AGENT, buy_icecream)))

Table 0.1 Formalisation of concepts involved in adaptation

The integration of the domain model relationships within a prediction model used for forward reasoning within the adaptation model can be done in a systematic manner by replacing the atoms in a domain model relationship, for example a & b ↠ c

by predictions and/or beliefs about them, thus obtaining:

belief(X, a) & prediction(X, b) ↠ prediction(X, c)

In this way the following prediction model was obtained by integration of the domain model:

PADR1 Generating beliefs from observations
If ambient agent *X* observes *Y*
then ambient agent *X* will believe *Y*
observed(X, Y) ↠ prediction(X, Y)

PADR2 Predicting that *A* becomes hungry
If ambient agent *X* believes that *A* does not eat
then ambient agent *X* will predict that *A* will be hungry
belief(X, not(performed(A, eat))) ↠ prediction(X, hungry(A))

PADR3 Incorporating an assumption on a parameter as a prediction
If ambient agent *X* assumes that *A* likes icecream
then ambient agent *X* predicts that *A* likes icecream
assumption(X, likes(A, icecream))
↠ prediction(X, likes(A, icecream))

PADR4 Predicting that *A* desires to eat
If ambient agent *X* predicts that *A* is hungry
and ambient agent *X* believes that *A* observes that at *P* they sell icecream
and ambient agent *X* predicts that *A* likes icecream
then ambient agent *X* will predict that *A* will desire to buy icecream
prediction(X, hungry(A)) &
belief(X, observed(A, icecream_at(P))) &
prediction(X, likes(A, icecream))
↠ prediction(X, desire(A, performed(A, buy_icecream)))

PADR5 Predicting that *A* goes to *P*
If ambient agent *X* predicts that *A* desires to buy icecream
and ambient agent *X* believes that *A* observes that at *P* they sell icecream
then ambient agent *X* will predict that *A* will go to *P*
prediction(X, desire(A, performed(A, buy_icecream))) &
belief(X, observed(A, icecream_at(P)))
↠ prediction(X, performed(A, goto(P)))

By comparing predictions and observations deviations are identified and based on them conclusions drawn on adaptation of parameter beliefs.

PADR6 Generating a deviation belief
If ambient agent *X* predicts that *A* goes to *P*
and ambient agent *X* observes that *A* does not go to *P*
then ambient agent *X* will believe that there is a deviation for *A* goes to *P*
prediction(X, performed(A, goto(P))) &
observed(X, not(performed(A, goto(P))))
↠ belief(X, has_deviation(performed(A, goto(P))))

PADR7 Generating an adapted parameter belief

If ambient agent *X* believes that there is a deviation for *A* goes to *P*
and ambient agent *X* assumed that *A* likes icecream
then ambient agent *X* will believe that *A* does not like icecream

belief(X, has_deviation(performs(A, goto(P)))) &
assumption(X, likes(A, icecream))
↠ belief(X, not(likes(A, icecream)))

5.3 An Example Trace for the Adaptation Model

In Figure 2.15, a trace is shown of a parameter adaptation process based on the model described in Section 0. From the assumption that *a* likes icecream it is predicted that *a* will have the desire to buy icecream and from that that *a* will go to *p1*. However, it is observed that *a* does not go to *p1*. Therefore a deviation is derived on going to *p1*. From this the belief is derived that *a* does not like icecream.

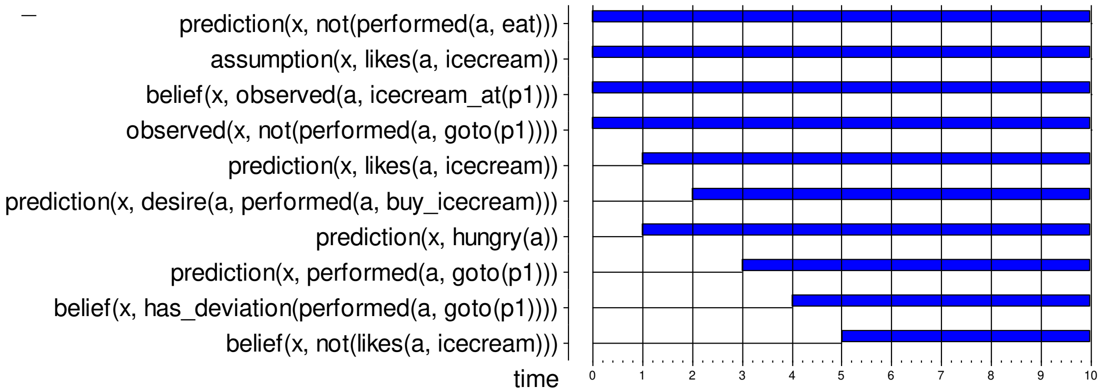

Figure 0.3 An example trace for a parameter belief adaptation process for the icecream case