IMPACT:
A Platform for Heterogenous Agents

Lecture Course given at
Ushuaia, Argentina

October 2000

Jürgen Dix,
University of Koblenz and Technical University of Vienna

1. ***IMPACT* Architecture**
2. **The Code Call Mechanism**
3. **Actions and Agent Programs**
4. **Regular Agents**
5. **Meta Agent Reasoning**
6. **Probabilistic Agent Reasoning**
7. **Temporal Agent Reasoning**

Based on the book

Heterogenous Active Agents
(Subrahmanian, Bonatti, Dix,
Eiter, Kraus, Özcan and Ross),
MIT Press, May 2000.

Timetable:

- 10 minutes to explain what is going on. Some sentences for each chapter.
- Chapter 1 can be entirely done in the remaining time.

7. Temporal Agent Reasoning

Overview

7.1 Timed Actions

7.2 Temporal Agent Programs

7.3 Semantics

Timetable:

- Chapter 7 needs 30 minutes.

7 Temporal Agent Reasoning

7.1 Timed Actions

- Most real-world actions have a duration. **heli**: **fly**("BA","US").
- It might be important to specify intermediate timepoints, **checkpoints** (Definition 7.1), and **to update the current state** incrementally at these prespecified points.

Thus, in order to specify a timed action, we must:

1. **Specify the total amount of time it takes for the action to be "completed".**
2. **Specify exactly how the state of the agent changes *while* the action is being executed.**

Definition 7.1 (Checkpoint Expressions $\mathsf{rel}:\{\mathtt{X} \mid \chi\}, \mathsf{abs}:\{\mathtt{X} \mid \chi\}$)

- *If $i \in \mathbb{N}$ is a positive integer, then $\mathsf{rel}:\{i\}$ and $\mathsf{abs}:\{i\}$ are checkpoint expressions.*
- *If χ is a code call condition involving a non-negative, integer-valued variable $\mathtt{X}$, then $\mathsf{rel}:\{\mathtt{X} \mid \chi\}$ and $\mathsf{abs}:\{\mathtt{X} \mid \chi\}$ are* checkpoint expressions.

Example 7.1
(Rescue: Checkpoints)

- $\mathsf{rel}:\{100\}$. *This says that a checkpoint occurs at the time of the start of the action, 100 units later, 200 units later, and so on.*
- $\mathsf{abs}:\{\mathtt{T} \mid \mathbf{in}(\mathtt{T}, \mathtt{clock}:\mathit{time}()) \,\&\, \mathbf{in}(\mathtt{0}, \mathtt{math}:\mathit{remainder}(\mathtt{T}, \mathtt{100})) \,\&\, \mathtt{T} > 5000\}$. *This says that a checkpoint occurs at absolute times 5000, 5100, 5200, and so on.*
- $\mathsf{abs}:\{\mathtt{T} \mid \mathbf{in}(\mathtt{T}, \mathtt{clock}:\mathit{time}()) \,\&\, \mathbf{in}(\mathtt{X}, \mathtt{a}:\mathit{getMessage}(\mathtt{comc})) \,\&\, \mathtt{X.Time} - \mathtt{T} = 5\}$. *This says that a checkpoint occurs at 5 time units after a message is received from the* $\mathtt{comc}$ *agent.*

Definition 7.2 (Timed Effect Triple $\langle\{chk\}, Add, Del\rangle$) A timed effect triple *is a triple of the form* $\langle\{chk\}, Add, Del\rangle$ *where* $\{chk\}$ *is a checkpoint expression, and Add and Del are add lists and delete lists.*

Example 7.2

(Rescue: Timed Effect Triples)

- *The* **truck** *agent may use the following timed effect triple to update its fuel at absolute times 5000, 5100, 5200, and so on.*

  1st arg :

  $\mathsf{abs}:\{\mathtt{T} \mid \mathbf{in}(\mathtt{T}, \mathsf{clock}:\mathit{time}()) \,\&\, \mathbf{in}(\mathtt{0}, \mathsf{math}:\mathit{remainder}(\mathtt{T}, \mathtt{100})) \,\&\, \mathtt{T} > 5000\}$

  2nd arg: $\{\mathbf{in}(\mathtt{NewFuelLevel}, \mathsf{truck}:\mathit{fuelLevel}(\mathtt{X}_{now}))\}$

  3rd arg : $\{\mathbf{in}(\mathtt{OldFuelLevel}, \mathsf{truck}:\mathit{fuelLevel}(\mathtt{X}_{now} - 20))\}$

Definition 7.3 (Timed Action)

A timed action α *consists of:*

Name: *A name, usually written* $\alpha(X_1,\ldots,X_n)$*, where the* X_i*'s are root variables.*

Schema: *A schema, usually written as* $(\tau_1,\ldots,\tau_n)$*, of types. Intuitively, this says that the variable* X_i *must be of type* τ_i*, for all* $1 \leq i \leq n$*.*

Pre: *A code-call condition* χ*, the* precondition *of the action, denoted by* $Pre(\alpha)$

Dur: *An expression of the form* $\{i\}$ *or* $\{\mathtt{X} \mid \chi\}$*. Depending on the current object state, this expression determines a* duration $\mathsf{duration}(\alpha) \in \mathbb{N}$ *of* α*.*

Tet: *A set* **Tet**(α) *of timed effect triples such that if both* $\langle\{chk\}, Add, Del\rangle$ *and* $\langle\{chk\}', Add', Del'\rangle$ *are in* **Tet**(α)*, then* $\{chk\}$ *and* $\{chk\}'$ *have no common solution w.r.t. any object state. The set* **Tet**(α) *together with* **Dur**(α) *determines the set of checkpoints* $\mathsf{checkpoints}(\alpha)$ *for action* α *(as defined below).*

Intuitively, if α is an action that we start executing at $t^{\alpha}_{\text{start}}$, then

- **Dur**(α) specifies how to compute the duration $\mathsf{duration}(\alpha)$ of α, and
- **Tet**(α) specifies the checkpoints associated with action α.

It is important to note that **Dur**(α) and **Tet**(α) may not specify the duration and checkpoint times explicitly (even if the associated checkpoints are of the form $\mathbf{abs}{:}\{\mathtt{X} \mid \chi\}$, i.e. absolute times). There is a method to compute $\mathsf{duration}(\alpha)$.

7.2 Temporal Agent Programs

Definition 7.4 (Temporal Annotation Item tai)

1. *Every integer is a temporal annotation item.*
2. *The distinguished integer valued variable* $\mathtt{X}_{now}$ *is a temporal annotation item.*
3. *Every integer valued variable is a temporal annotation item.*
4. *If* $tai_1, \ldots, tai_n$ *are temporal annotation items, and* $b_1, \ldots, b_n$ *are integers (positive or negative), then* $(b_1 tai_1 + \ldots + b_n tai_n)$ *is a temporal annotation item.*

- 1, $\mathtt{X}_{\mathrm{now}}$, $\mathtt{X}_{\mathrm{now}} + 3$, $\mathtt{X}_{\mathrm{now}} + 2v + 4$ are all temporal annotation items if v is an integer valued variable.
- Temporal annotation items, when ground, evaluate to time points. They are used to specify a time interval.

Definition 7.5 (Temporal Annotation $[\mathsf{tai}_1,\mathsf{tai}_2]$)
If $\mathsf{tai}_1, \mathsf{tai}_2$ *are annotation items, then* $[\mathsf{tai}_1,\mathsf{tai}_2]$ *is a temporal annotation.*

- $[2,5]$ is a temporal annotation item describing the set of time points between 2 and 5 (inclusive).
- $[2,3\mathtt{X}+4\mathtt{Y}]$ is a temporal annotation.
- When $\mathtt{X} := 2, \mathtt{Y} := 3$, this defines the set of time points between 2 and 18. $[\mathtt{X}_{\mathsf{now}},\mathtt{X}_{\mathsf{now}}+5]$ is a temporal annotation.

Definition 7.6 ((Temporal) Action State Condition)
Suppose χ *is a (possibly empty) code call condition,* $L_1,\dots,L_n$ *are action status literals, and* `ta` *is a temporal annotation. Then:*

1. $(\chi \& L_1 \& \dots \& L_n)$ *is called an* action state condition.
2. $(\chi \& L_1 \& \dots \& L_n) : \mathtt{ta}$ *is called a* temporal action state conjunct *(tasc).*
3. *If* χ *is empty, then* $(L_1 \& \dots \& L_n) : \mathtt{ta}$ *is called a* state-independent tasc .

Intuitively, when $\rho : \mathtt{ta}$ is ground for some action state condition ρ, we may read this as "ρ is true at some point in `ta`". The following is a simple tasc.

- $(\mathbf{in}(\mathtt{X}, \mathbf{heli} : \mathbf{\mathit{inventory}}(\mathtt{fuel})) \& \mathtt{X.Qty} < 50) : [\mathtt{X}_{\mathsf{now}} - 10, \mathtt{X}_{\mathsf{now}}]$.
  Intuitively, this tasc is true if at some point in time t_i in the last 10 time units, the helicopter had less than 50 gallons of fuel left.

We are now ready to define the most important syntactic construct of this chapter, a *temporal agent rule*.

Definition 7.7 (Temporal Agent Rule/Program $\mathcal{TP}$)
A temporal agent rule *is an expression of the form*

$$Op\,\alpha : [tai_1, tai_2] \leftarrow \rho_1 : \mathtt{ta}_1 \,\&\cdots\&\, \rho_n : \mathtt{ta}_n,$$

where $Op \in \{\mathbf{P}, \mathbf{Do}, \mathbf{F}, \mathbf{O}, \mathbf{W}\}$*, and* $\rho_1 : \mathtt{ta}_1, \ldots, \rho_n : \mathtt{ta}_n$ *are tascs.*

A temporal agent program *is a finite set of temporal agent rules.*

Intuitive Reading of Temporal Agent Rule

“If for all $1 \leq i \leq n$*, there exists a time point* t_i *such that* ρ_i *is true at time* t_i *such that either*

1. ρ_i *is state independent and* $t_i \in \mathtt{ta}_i$*, or*
2. ρ_i *is not state independent and* $t_i \leq \mathtt{t}_{now}$ *(i.e.* t_i *is now or is in the past) and* $t_i \in \mathtt{ta}_i$*,*

then $\mathsf{Op}\alpha$ *is true at some point* $t \geq \mathtt{t}_{now}$ *(i.e. now or in the future) such that* $tai_1 \leq t \leq tai_2$*”.*

$$\mathsf{Op}\alpha : [\mathsf{tai}_1,\mathsf{tai}_2] \leftarrow \rho_1 : \mathtt{ta}_1 \,\& \cdots \&\, \rho_n : \mathtt{ta}_n,$$

“If a prediction package expects a stock to rise K% after T_K units of time and $K \geq 25$ then buy the stock at time $(\mathtt{X}_{\mathsf{now}} + T_K - 2)$.”

We assume a prediction package that given a stock uses (some stock expertise) to predict the change in the value of the stock at future time points. This function returns a set of pairs of the form (T, C). Intuitively, this says that T units from now, the stock price will change by C percent (positive or negative).

$$\mathbf{Do}\,\mathit{buy}S : [\mathtt{X}_{\mathsf{now}} + \mathtt{X.T} - 2, \mathtt{X}_{\mathsf{now}} + \mathtt{X.T} - 2] \leftarrow$$
$$(\mathbf{in}(\mathtt{X}, \mathtt{pred} : \mathit{dest}(\mathtt{S})) \,\&\, \mathtt{X.C} \geq 25) : [\mathtt{X}_{\mathsf{now}}, \mathtt{X}_{\mathsf{now}}].$$

7.3 Semantics

Definition 7.8 (Temporal Status Set $\mathcal{TS}_{\mathtt{t}_{\mathbf{now}}}$)
A temporal status set $\mathcal{TS}_{\mathtt{t}_{now}}$ *at time* $\mathtt{t}_{now}$ *is a mapping from natural numbers to ordinary status sets satisfying* $\mathcal{TS}_{\mathtt{t}_{now}}(i) = \emptyset$ *for all* $i > i_0$ *for some* $i_0 \in \mathbb{N}$.

As usual a feasible status set must satisfy

- Closure under rules,
- Deontic consistency wrt. **State History** (⇝ Definition 7.9).
- Deontic closure,
- **Checkpoint consistency** (⇝ Definition 7.10).

As an agent that reasons about time may need to reason about the current, as well as past states it was/is in, a notion of state history is needed.

Definition 7.9 (State History Function $\mathsf{hist}_{\mathtt{t}_{\mathsf{now}}}$)
A state history function $\mathit{hist}_{\mathtt{t}_{now}}$ *at time* $\mathtt{t}_{now}$ *is a partial function from* $\mathbb{N}$ *to agent states such that* $\mathit{hist}_{\mathtt{t}_{now}}(\mathtt{t}_{now})$ *is always defined and for all* $i > \mathtt{t}_{now}$, $\mathit{hist}_{\mathtt{t}_{now}}(i)$ *is undefined.*

The definition of state history does not *require* that an agent store the entire past.

1. He may decide to store no past information at all. In this case, $\mathsf{hist}_{\mathsf{t}_{\mathsf{now}}}(i)$ is defined *if and only if* $i = \mathsf{t}_{\mathsf{now}}$.
2. He may decide to store information only about the past i units of time. This means that he stores the agent's state at times $\mathsf{t}_{\mathsf{now}}, (\mathsf{t}_{\mathsf{now}} - 1), \ldots, (\mathsf{t}_{\mathsf{now}} - i)$, i. e. $\mathsf{hist}_{\mathsf{t}_{\mathsf{now}}}$ is defined for the following arguments: $\mathsf{hist}_{\mathsf{t}_{\mathsf{now}}}(\mathsf{t}_{\mathsf{now}}), \mathsf{hist}_{\mathsf{t}_{\mathsf{now}}}(\mathsf{t}_{\mathsf{now}} - 1), \ldots, \mathsf{hist}_{\mathsf{t}_{\mathsf{now}}}(\mathsf{t}_{\mathsf{now}} - i)$ are defined.
3. He may decide to store, in addition to the current state, the history every five time units. That is, $\mathsf{hist}_{\mathsf{t}_{\mathsf{now}}}(\mathsf{t}_{\mathsf{now}})$ is defined and for each $0 \leq i \leq \mathsf{t}_{\mathsf{now}}$, if $i \mod 5 = 0$, then $\mathsf{hist}_{\mathsf{t}_{\mathsf{now}}}(i)$ is defined. Such an agent may be specified by an agent designer when he believes that maintaining some (but not all) past snapshots is adequate for his application's needs.

For a temporal status set to be feasible, at each checkpoint the state needs to be updated.
The expected future states of the agent need to satisfy the integrity constraints.

Definition 7.10 (Checkpoint Consistency)

$\mathcal{TS}_{\mathtt{t}_{now}}$ *is said to be* checkpoint consistent *at time* $\mathtt{t}_{now}$ *if, by definition, for all* $i > \mathtt{t}_{now}$, $\mathcal{EO}(i)$ *(see Definition 7.11) satisfies the integrity constraints* $\mathcal{IC}$.

Definition 7.11 (Expected States at time $\mathtt{t}$: $\mathcal{EO}(\mathtt{t})$)
Suppose the current time is $\mathtt{t}_{now}$, $hist_{\mathtt{t}_{now}}$ *is the agent's state history function and* $\mathcal{TS}_{\mathtt{t}_{now}}$ *is a temporal status set. The agent's expected states are defined as follows:*

- $\mathcal{EO}(\mathtt{t}_{now}) = hist_{\mathtt{t}_{now}}(\mathtt{t}_{now})$.
- *For all time points* $i > \mathtt{t}_{now}$, $\mathcal{EO}(i)$ *is the result of concurrently executing*

  $\{\alpha \mid \mathbf{Do}\,\alpha \in TS_{now}(i-1)\} \cup$
  $\{\beta' \mid \mathbf{Do}\,\beta \in TS_{now}(j)$ *for* $j \leq i-1$ *and* $i-1$ *is a checkpoint for* β, *and* β'
  denotes the action (non-timed) which has an empty precondition,
  and whose add and del lists are as specified by $\mathbf{Tet}(\beta)\}$

  in state $\mathcal{EO}(i-1)$.

We note that that from a certain $i_0 \in \mathbb{N}$ *onwards, we have* $\mathcal{EO}(i) = \emptyset$ *for all* $i > i_0$ *(this is because of the same property for the action history and the temporal status set).*

References

Apt, K., H. Blair, and A. Walker (1988). Towards a Theory of Declarative Knowledge. In J. Minker (Ed.), *Foundations of Deductive Databases and Logic Programming*, pp. 89–148. Washington DC: Morgan Kaufmann.

Arens, Y., C. Y. Chee, C.-N. Hsu, and C. Knoblock (1993). Retrieving and Integrating Data From Multiple Information Sources. *International Journal of Intelligent Cooperative Information Systems 2*(2), 127–158.

Arisha, K., F. Ozcan, R. Ross, V. S. Subrahmanian, T. Eiter, and S. Kraus (1999, March/April). IMPACT: A Platform for Collaborating Agents. *IEEE Intelligent Systems 14*, 64–72.

Bayardo, R., et al. (1997). Infosleuth: Agent-based Semantic Integration of Information in Open and Dynamic Environments. In J. Peckham (Ed.), *Proceedings of ACM SIGMOD Conference on Management of Data*, Tucson, Arizona, pp. 195–206.

Brink, A., S. Marcus, and V. Subrahmanian (1995). Heterogeneous Multimedia Reasoning. *IEEE Computer 28*(9), 33–39.

Chawathe, S., et al. (1994, October). The TSIMMIS Project: Integration of Heterogeneous Information Sources. In *Proceedings of the 10th Meeting of the Information Processing Society of Japan*, Tokyo, Japan. Also available via anonymous FTP from host db.stanford.edu, file /pub/chawathe/1994/tsimmis-overview.ps.

Dix, J., S. Kraus, and V. Subrahmanian (2001). Temporal agent reasoning. *Artificial Intelligence to appear*.

Dix, J., M. Nanni, and V. S. Subrahmanian (2000). Probabilistic agent reasoning. *Transactions of Computational Logic 1*(2).

Dix, J., V. S. Subrahmanian, and G. Pick (2000). Meta Agent Programs. *Journal of Logic Programming 46*(1-2), 1–60.

Eiter, T., V. Subrahmanian, and T. J. Rogers (2000). Heterogeneous Active Agents, III: Polynomially Implementable Agents. *Artificial Intelligence 117*(1), 107–167.

Eiter, T. and V. S. Subrahmanian (1999). Heterogeneous Active Agents, II: Algorithms and Complexity. *Artificial Intelligence 108*(1-2), 257–307.

Genesereth, M. R. and S. P. Ketchpel (1994). Software Agents. *Communications of the ACM 37*(7), 49–53.

Rogers Jr., H. (1967). *Theory of Recursive Functions and Effective Computability*. New York: McGraw-Hill.

Subrahmanian, V., P. Bonatti, J. Dix, T. Eiter, S. Kraus, F. Özcan, and R. Ross (2000). *Heterogenous Active Agents*. MIT-Press.

Wiederhold, G. (1993). Intelligent Integration of Information. In *Proceedings of ACM SIGMOD Conference on Management of Data*, Washington, DC, pp. 434–437.

Wilder, F. (1993). *A Guide to the TCP/IP Protocol Suite*. Artech House.