

Corso di Interazione Uomo-Macchina I

Paolo Bottoni

DIPARTIMENTO
DI INFORMATICA



SAPIENZA
UNIVERSITÀ DI ROMA

Lezione 6: Analisi dei compiti

Lucidi tradotti e adattati dal materiale presente sul sito <http://www.hcibook.com/e3/resources/>

Analisi dei compiti

Metodi per analizzare lavori persone:

- Cosa fanno
- Con quali cose lavorano
- Cosa devono sapere

Gerarchie di obiettivi e compiti

- Astrazione dai processi interni dell'utente
- Elaborazione mentale come dividi-e-conquista
- Esempio: produrre rapporto vendite
 - produce report
 - gather data
 - . find book names
 - . . do keywords search of names database
 - *further sub-goals*
 - . . sift through names and abstracts by hand
 - *further sub-goals*
 - . search sales database - further sub-goals
 - layout tables and histograms - further sub-goals
 - write description - further sub-goals
- Necessarie conoscenze su:
 - Dati, nomi, vendite, tabelle, etc.

Metodo generale

- Osservare
- Riferimenti a procedure formali e informali
- Raccogliere liste non strutturate di parole e azioni
- Organizzare usando notazioni o diagrammi

Obiettivi dell'analisi dei compiti

- Identificare azioni / competenze / procedure necessarie utente per svolgere attività
- Azioni tipiche in contesto di lavoro
- NON sono descrizione di come usare sistema da progettare
 - Analisi preliminare a progetto
- Sintassi linguaggi usati per analisi compiti può essere utilizzata per descrivere uso

Differenze da altre tecniche

Analisi di sistema

progetto di sistema

vs.

Analisi dei compiti

- focus - utente

Modelli cognitivi

stato mentale interno

vs.

Analisi dei compiti

- focus - azioni esterne

compito "unitario" praticato

- focus - intero lavoro

Fonti di informazione

Documentazione

- N.B. manuali dicono cosa *dovrebbe* avvenire, buoni per parole chiave e stimolare interviste

Osservazione

- formale/informale, laboratorio/campo

Interviste

- esperto: dirigente o lavoratore? (chiedere a entrambi!)

Analisi iniziale

Estrarre da trascrizioni

- elencare nomi (oggetti) e verbi (azioni)
- Attenzione a linguaggio tecnico e contesto:
`the rain poured` vs. `I poured the tea`

Ordinare e classificare

- Raggruppare o disporre parole su schede
- Ordinare oggetti / azioni per rilevanza a compito
- Usare strumenti per schematizzazione

Processo iterativo:

fonti di dati  analisi

... ma costoso, usare fonti economiche quando disponibili

Usi – requisiti & progetto

Cattura requisiti e progetto di sistema

- Alza fuoco da sistema a uso
- Suggerisce candidati per automazione
- Rivela modello concettuale utente

Progetto dettagliato di interfaccia

- Tassonomie suggeriscono organizzazione menu
- Elenchi oggetti/azione suggeriscono oggetti di interfaccia
- Frequenza compiti guida scelte di default
- Sequenze di compiti esistenti guidano progetto dialogo

NOTA. Analisi compiti non è mai completa

- Progetto basato su compiti rigidi \Rightarrow sistema inflessibile

Usi – manuali & documentazione

Manuale Concettuale

- Da analisi basata su conoscenza o entità-relazione
- Buono per compiti aperti

Manuale Procedurale ‘Come si fa’

- Da descrizione HTA
- Buono per principianti
- Assume tutti compiti noti

To make cups of tea

boil water — see page 2
empty pot
make pot — see page 3
wait 4 or 5 minutes
pour tea — see page 4

— page 1 —

Make pot of tea *once water has boiled*

warm pot
put tea leaves in pot
pour in boiling water

— page 3 —

Approcci ad analisi dei compiti

- Decomposizione compiti
 - Sottocompiti ordinati temporalmente e causalmente
- Tecniche basate su conoscenza
 - Conoscenze utente su compito e sua organizzazione
- Analisi basata su entità/oggetti
 - Relazioni tra oggetti, azioni e persone che le svolgono

Decomposizione in compiti

Finalità:

- Descrivere azioni svolte da persone che devono interagire
- Strutturarle in gerarchia compiti e sotto compiti
- Descrivere ordine sottocompiti

Varianti:

- Hierarchical Task Analysis (HTA)
 - più diffusa
- CTT (CNUCE, Pisa)
 - usa operatori temporali di LOTOS

Descrizione testuale HTA

Descrizione della gerarchia ...

0. per pulire casa
 1. prendere aspirapolvere
 2. trovare accessori appropriati
 3. pulire stanze
 - 3.1. pulire sala
 - 3.2. pulire stanze da letto
 - 3.3. pulire bagni
 4. vuotare sacchetto polvere
 5. rimettere via aspirapolvere e estensioni

... e dei piani

Piano 0: svolgi 1 - 2 - 3 - 5 in ordine, quando sacchetto si riempie esegui 4

Piano 3: fai uno di 3.1, 3.2 or 3.3 in qualsiasi ordine, in base a stanze da pulire

N.B. solo piani denotano ordine

Generazione della gerarchia

- 1 Ottieni lista dei compiti
- 2 Raggruppa compiti in compiti di livello più alto
- 3 Decomponi ulteriormente compiti di livello più basso

Regole di stop

Come si sa quando fermarsi?

Compito come “svuota sacchetto” abbastanza semplice?

Scopo: espandere solo compiti rilevanti

Azioni motorie: minimo livello ragionevole

Compiti come spiegazioni

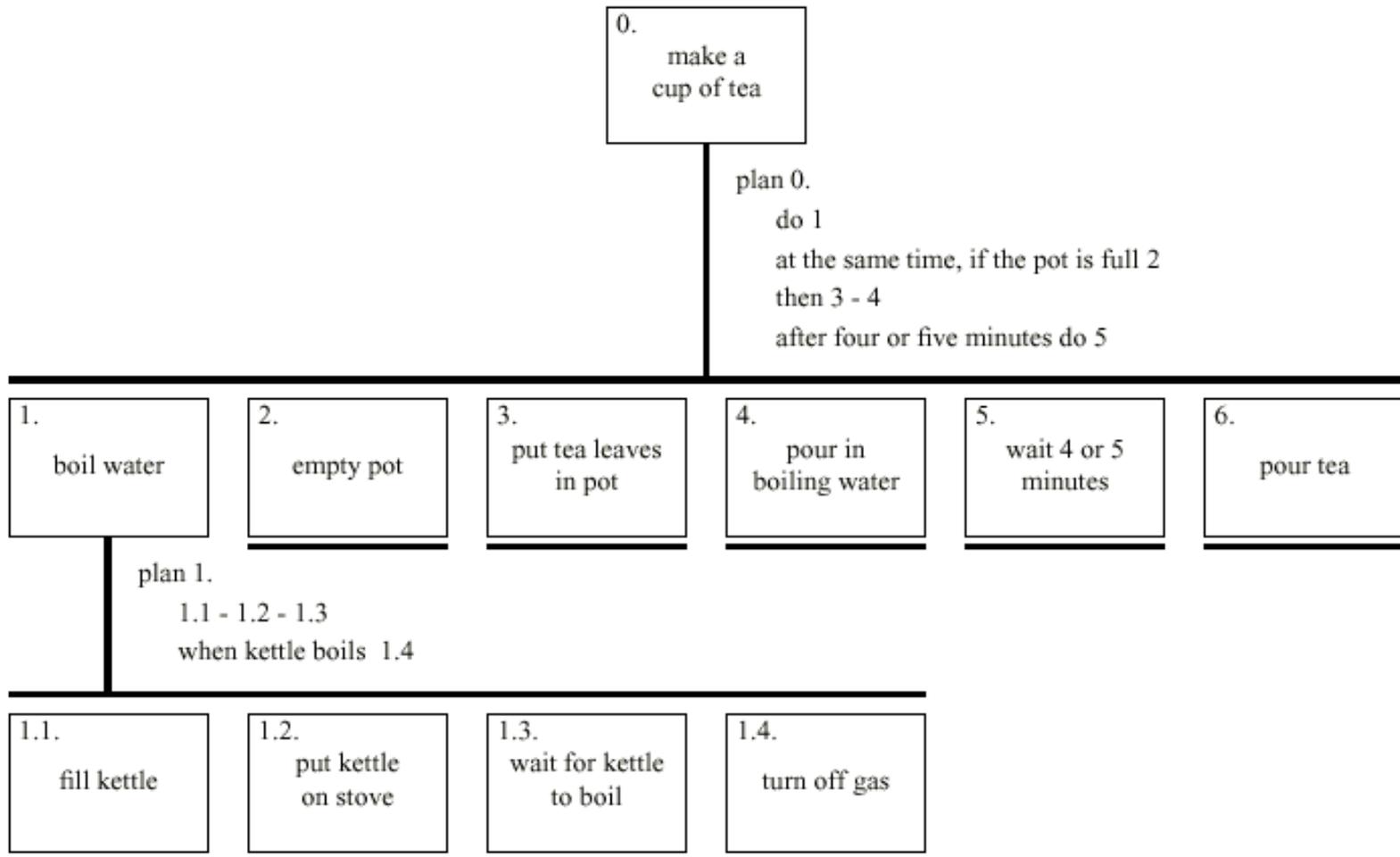
- Domanda:
cosa sta facendo?
- Livelli di spiegazione:
digito ctrl-B
rendo parola in grassetto
enfattizzo parola
edito documento
scrivo lettera
preparo caso legale

Scenario di analisi usando HTA



0. in order to clean the house
 1. get the vacuum cleaner out
 2. get the appropriate attachment
 3. clean the rooms
 - 3.1. clean the hall
 - 3.2. clean the living rooms
 - 3.3. clean the bedrooms
 4. empty the dust bag
 5. put vacuum cleaner and attachments away

HTA diagrammatica



Raffinamento delle descrizioni

Data HTA (testuale o diagrammatica)

Come verificarla / migliorarla?

Euristiche:

Azioni accoppiate

e.g., manca `turn on gas'

Ristrutturare

e.g., generare compito `make pot'

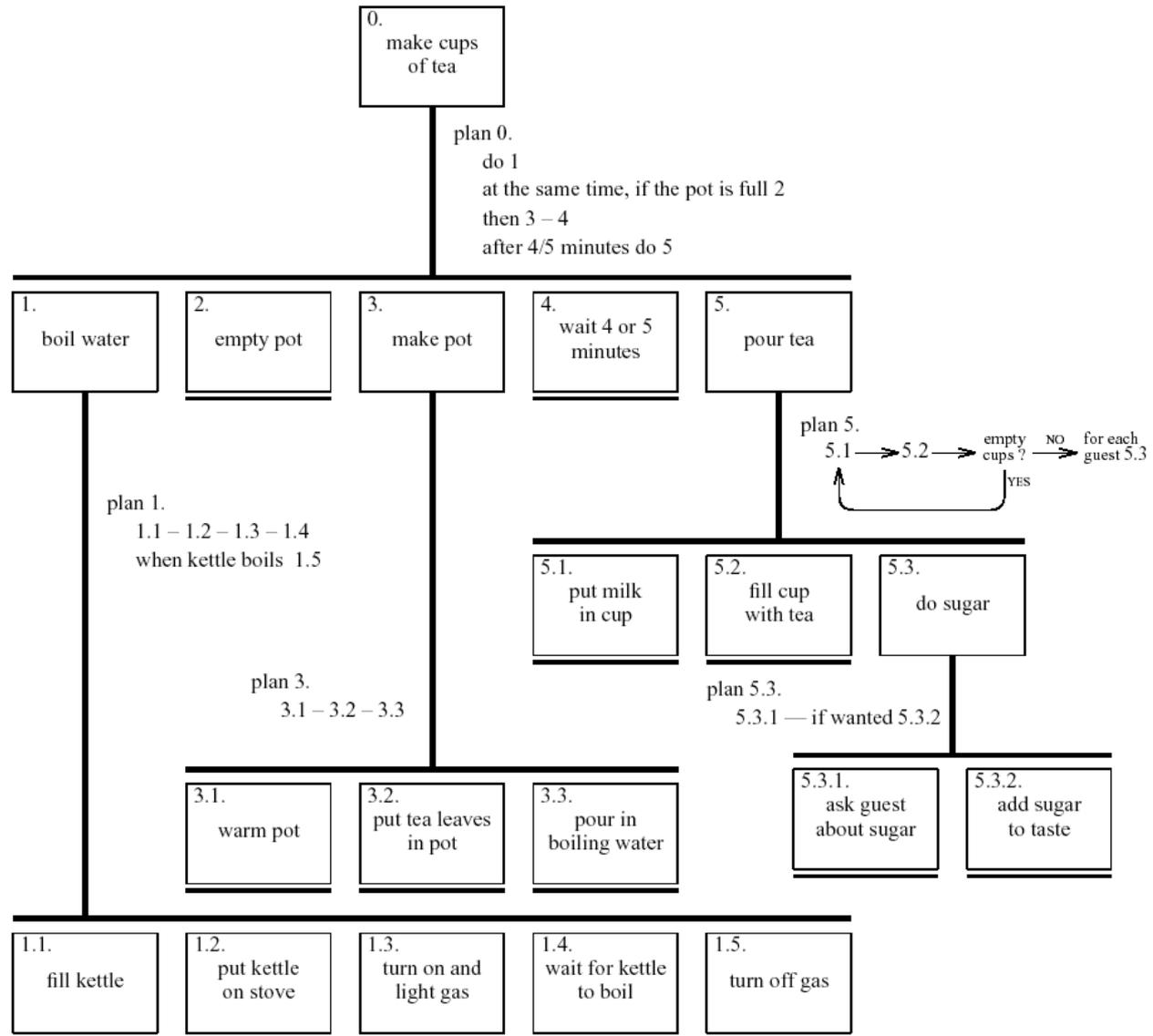
Equilibrare

e.g. `pour tea' più semplice di “make pot”?

Generalizzare

e.g., “make a cup or more”

HTA diagrammatico raffinato

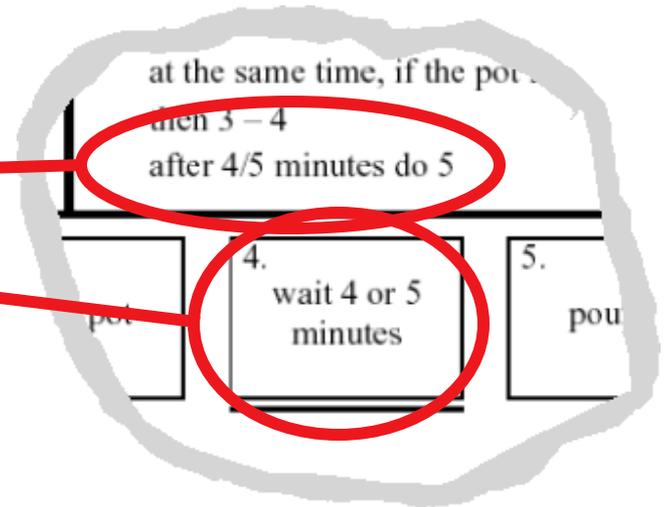


Tipi di piano

- Sequenza fissata - 1.1 then 1.2 then 1.3
- Compiti opzionali - if the pot is full 2
- Attesa di eventi - when kettle boils 1.4
- Cicli - do 5.1 5.2 while there are still empty cups
- Concorrenti - do 1; at the same time ...
- Discrezionali - do any of 3.1, 3.2 or 3.3 in any order
- Mescolanze - in genere un piano presenta vari costrutti

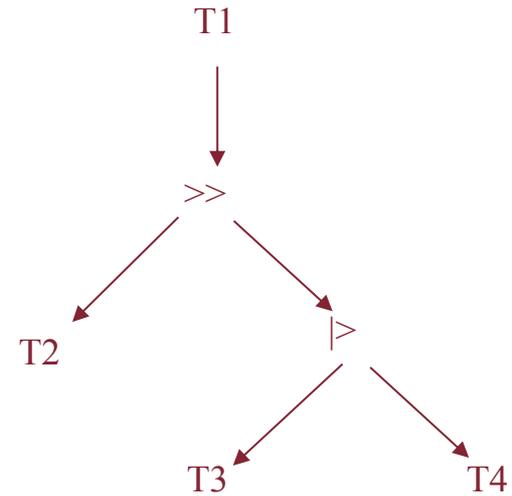
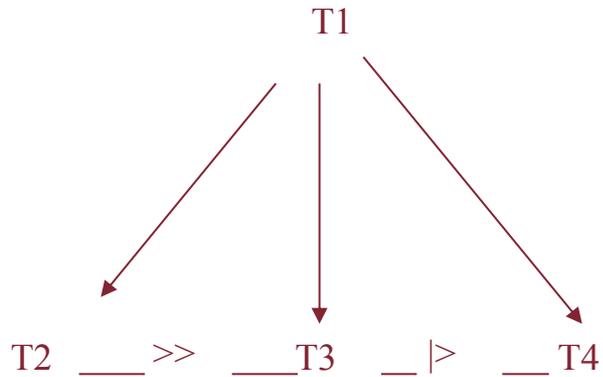
Forme di attesa ...

- Parte di piano?
... o di compito?
- Generalmente
 - Di compito – se ‘busy’ wait
 - Si attende attivamente
 - Di piano – se fine del ritardo è evento
 - e.g. “when alarm rings”, “when reply arrives”

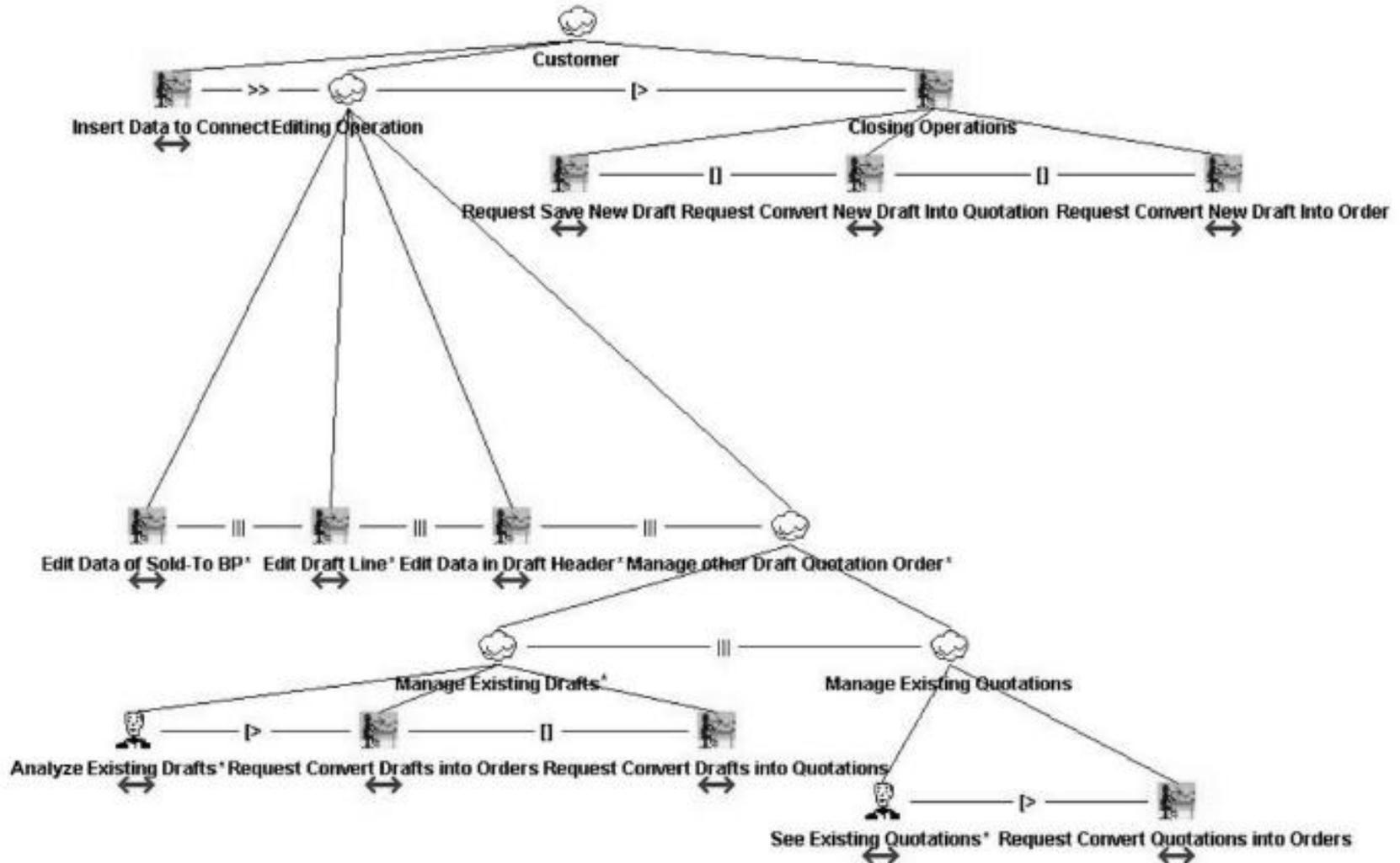


CTT

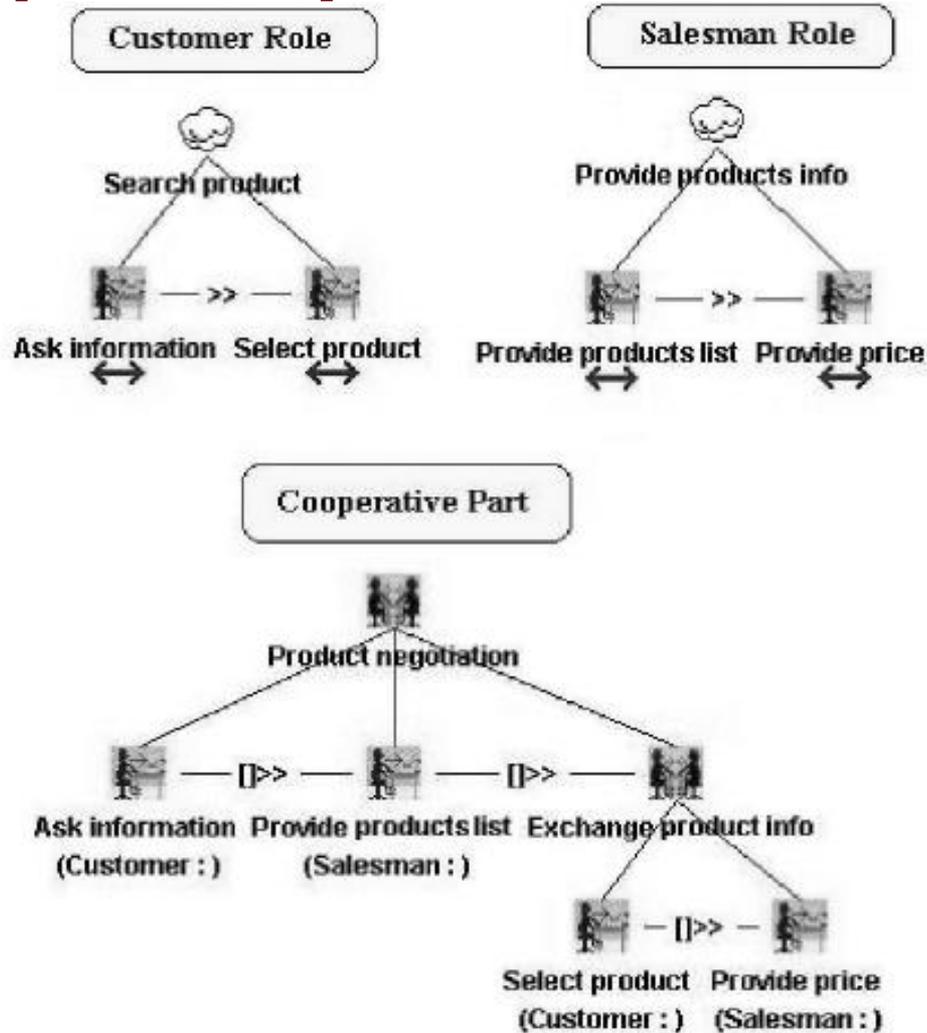
- Operatori logico-temporali



Un compito complesso



Compiti cooperativi



Operatori I

- $T1 \parallel T2$ Concorrenza indipendente
- $T1 [] T2$ Scelta
- $T1 |[[]| T2$ Concorrenza con scambio di informazioni
- $T1 [> T2$ Disattivazione
- $T1 >> T2$ Abilitazione

Operatori II

- $T1 \triangleright T2$ Abilitazione con passaggio di informazioni
- $T1 \mid \triangleright T2$ Sospendi / riprendi
- T^* Iterazione
- $T1(n)$ Iterazione finita
- $[T]$ Task opzionale
Ricorsione

Analisi basate su conoscenza

Fuoco su:

Oggetti – usati in compito

Azioni – eseguite

Tassonomie – rappresentano livelli astrazione

Esempio basato su conoscenza

Controlli auto

guida *volante, frecce*

motore/velocità

dirette *accensione, acceleratore, pedale freno*

marce *frizione, cambio*

luci

esterne *fari, luci di pericolo*

interne *luci di cortesia*

lavare/tergere

tergicristalli *anteriori, posteriori*

spruzzi *anteriori, posteriori*

riscaldamento

controllo temperatura, direzione aria, ventilatore, scaldatore vetro posteriore

parcheggiamento *freno a mano, fermo porta*

radio *numerosi!*

Gerarchia di descrizione di compiti

Tre tipi di ramificazioni nella tassonomia:

- XOR – tassonomia normale
oggetto in uno e un solo ramo
- AND – oggetto deve essere in entrambi
classificazioni multiple
- OR – caso più debole
può essere in uno, nessuno o molti

```
wash/wipe AND
  function XOR
    wipe      front wipers, rear wipers
    wash      front washers, rear
washers
  position XOR
    front     front wipers, front
washers
    rear      rear wipers, rear washers
```

Esempio TDH

```
kitchen item AND
/___shape XOR
  |___dished  mixing bowl, casserole, saucepan,
  |           soup bowl, glass
  |___flat    plate, chopping board, frying pan
/___function OR
  {___preparation  mixing bowl, plate, chopping board
  {___cooking      frying pan, casserole, saucepan
  {___dining XOR
    |___for food   plate, soup bowl, casserole
    |___for drink  glass
```

N.B. ‘/ | {’ usati per tipi di ramificazione.

Ancora su TDH

Regola di unicità:

- diagramma può distinguere ogni oggetto?

es., piatto è:

```
kitchen item/shape(flat)/function{preparation,dining(for food)}/
```

nessun altro oggetto corrisponde a questa descrizione

Anche azioni hanno tassonomia:

```
kitchen job OR
```

```
|___ preparation beating, mixing
```

```
|___ cooking frying, boiling, baking
```

```
|___ dining pouring, eating, drinking
```

Astrazione e tagli

Dopo avere prodotto tassonomia dettagliata
"taglio" per produrre vista astratta

Ignora nodi a livelli inferiori

es. tagliando ramo *shape* e sottoalbero di *dining, plate* diventa:

```
kitchen item/function{preparation,dining}/
```

Termine in Knowledge Representation Grammar (KRG)

Possono essere più complesse:

es. 'beating in a mixing bowl' diventa:

```
kitchen job(preparation) using a  
kitchen item/function{preparation}/
```

Tecniche Entità-Relazioni

Fuoco su oggetti, azioni, e loro relazioni

Simile a analisi OO, ma ...

- Include entità non da calcolatore
- Enfattizza comprensione dominio, non implementazione

esempio

‘Vera's Veggies’ – ditta di orticoltura

Proprietario/gestore: Vera Bradshaw

Impiegati: Sam Gummage e Tony Peagreen

Vari strumenti incluso trattore ‘Fergie’

2 campi e 1 serra

Nuovo sistema di irrigazione controllato da calcolatore

Oggetti

Iniziare con lista di oggetti e classificarli:

Oggetti concreti:

Cose semplici: vanga, aratro, serra

Attori:

Attori umani: Vera, Sam, Tony, clienti

Controllore irrigazione?

Oggetti composti:

insiemi: squadra = Vera, Sam, Tony

tuple: trattore può essere < Fergie, aratro >

Attributi

Aggiungere attributi a oggetti:

Object Pump3 simple – irrigation pump

Attributes:

status: on/off/faulty

capacity: 100 litres/minute

N.B. non necessita completezza computazionale

Azioni I

Elencare azioni e associare a ciascuna:

agente – che esegue azione

paziente – che è modificato da azione

strumento – usato per eseguire azione

esempi:

Sam (*agent*) planted (*action*) the leeks (*patient*)

Tony dug the field *with* the spade (*instrument*)

Azioni II

Agenti impliciti – leggere dietro parole

`the field was ploughed' – *da chi?*

Agente indiretto – vero agente?

`Vera programmed the *controller* to irrigate the field'

Messaggi – tipo speciale di azione

`Vera *told* Sam to ... '

Ruoli – agente agisce in diversi ruoli

Vera as *worker* or as *manager*

Esempio – oggetti e azioni

Object Sam human actor

Actions:

S1: drive tractor

S2: dig the carrots

Object Vera human actor

– the proprietor

Actions: as worker

V1: plant marrow seed

V2: program irrigation controller

Actions: as manager

V3: tell Sam to dig the carrots

Object the men composite

Comprises: Sam, Tony

Object glasshouse simple

Attribute:

humidity: 0-100%

Object Irrigation Controller non-human actor

Actions:

IC1: turn on Pump1

IC2: turn on Pump2

IC3: turn on Pump3

Object Marrow simple

Actions:

M1: germinate

M2: grow

Eventi

... quando accade qualcosa

- Esecuzione di azione
‘Sam dug the carrots’
- Eventi spontanei
‘the marrow seed germinated’
‘the humidity drops below 25%’
- Eventi temporizzati
‘at midnight the controller turns on’

Relazioni

- Oggetto-oggetto
 - sociale - Sam è subordinato a Vera
 - spaziale - pompa 3 è in serra
- Azione-oggetto
 - agente (elencato con oggetto)
 - paziente e strumento
- Azioni e eventi
 - temporali e causali
 - 'Sam ha piantato le carote perché Vera gli ha detto di farlo''
- Relazioni temporali
 - Usare HTA o notazioni di dialogo.
 - Mostrare sequenze azioni (HTA normali)
 - Mostrare ciclo di vita oggetti

Esempi – eventi e relazioni

Events:

Ev1: humidity drops below 25%

Ev2: midnight

Relations: object-object

location (Pump3, glasshouse)

location (Pump1, Parker's Patch)

Relations: action-object

patient (V3, Sam)

- Vera tells *Sam* to dig

patient (S2, the carrots)

- Sam digs the *carrots* ...

instrument (S2, spade)

- ... *with* the spade

Relations: action-event

before (V1, M1)

- the marrow must be sown *before* it can germinate

triggers (Ev1, IC3)

- *when* humidity drops below 25%, the controller turns on pump 3

causes (V2, IC1)

- the controller turns on the pump *because* Vera programmed it