

Corso di Interazione Uomo-Macchina

Paolo Bottoni

DIPARTIMENTO
DI INFORMATICA



SAPIENZA
UNIVERSITÀ DI ROMA

Lezione 14: Notazioni e progetto del dialogo

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

Notazioni e progetto del dialogo

- Notazioni per dialogo
 - Diagrammatiche
 - Reti di transizioni di stato, diagrammi di flusso
 - Testuali
 - Grammatiche formali, regole di produzione, CSP
- Dialogo connesso a:
 - Semantica sistema – cosa fa
 - Presentazione sistema – come **appare**
- Descrizioni formali possono essere analizzate per trovare
 - Azioni incoerenti
 - Azioni difficili da invertire
 - Azioni mancanti
 - Potenziali errori di battitura

Stili di interazione

Dialogo ... calcolatore e utente

Diversi stili di interazione

Cos'è il dialogo?

- Conversazione tra due o più parti
 - Solitamente cooperativa
- In interfacce utente
 - Si riferisce a struttura interazione
 - Livello sintattico “conversazione” umano-calcolatore
- Livelli
 - Lessicale – forme icone, tasti effettivamente premuti
 - Sintattico – ordine di ingressi e uscite
 - Semantico – effetto su dati interni applicazione

Dialogo umano strutturato

- Dialogo umano-calcolatore molto vincolato
- Alcuni dialoghi umano-umano pure formali

Minister: do you *man's name* take this woman ...

Man: I do

Minister: do you *woman's name* take this man ...

Woman: I do

Man: With this ring I thee wed

(places ring on womans finger)

Woman: With this ring I thee wed *(places ring ..)*

Minister: I now pronounce you man and wife

Lezioni sul dialogo

- Cerimonia di matrimonio
 - Tipo di copione per tre partecipanti
 - Specifica di ordine
 - Alcuni contributi fissi – “I do”
 - Altri variabili - “do you *man’s name* ...”
 - Istruzioni per anello concorrenti con pronunciare parole “with this ring ...”
- Se si dicono queste parole si è sposati?
 - Solo se posto giusto, con licenza di matrimonio
 - Sintassi non semantica

... e inoltre

- Cosa succede se qualcuno dice “I don’t”?
- Dialoghi reali spesso hanno alternative:

Judge: How do you plead guilty or not guilty?

Defendant: *either* Guilty *or* Not guilty

- Svolgimento processo dipende da risposta imputato
- Fuoco su risposte normative:
 - Non considera fuoruscite da schema
 - Nel caso di GUI, azioni fisiche utente irrilevanti
 - Ma non per interazione multimodale!

Notazioni per progetto del dialogo

- Dialogo sepolto in programma
- In grande sistema è possibile?
 - Analizzare dialogo
 - Può utente controllare sempre proprio cestino acquisti?
 - Cambiare piattaforma
 - (es. Windows/Mac)
 - Notazioni di dialogo aiutano a:
 - Analizzare sistemi
 - Separare aspetti lessicali da semantici
- ... e prima che sistema sia costruito
 - Notazioni aiutano a comprendere progetti proposti

Notazioni grafiche

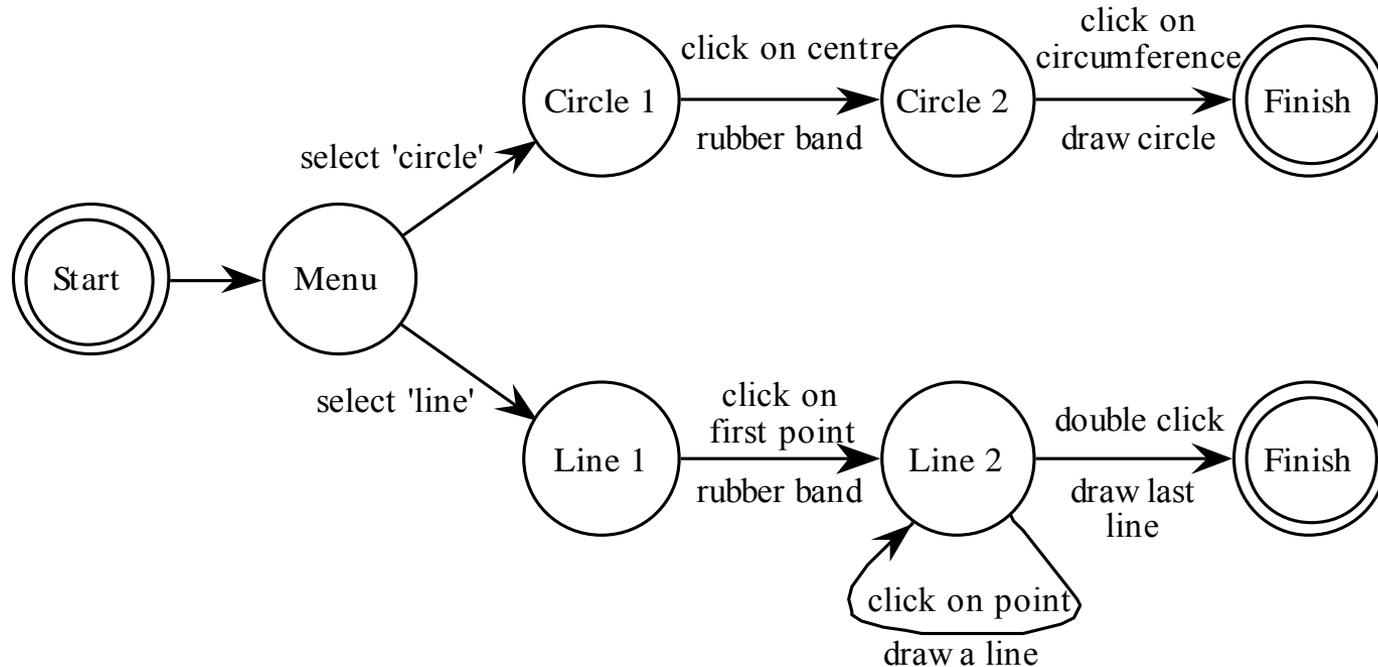
Reti di transizioni di stato (STN)

Reti di Petri, diagrammi di stato

Diagrammi di flusso

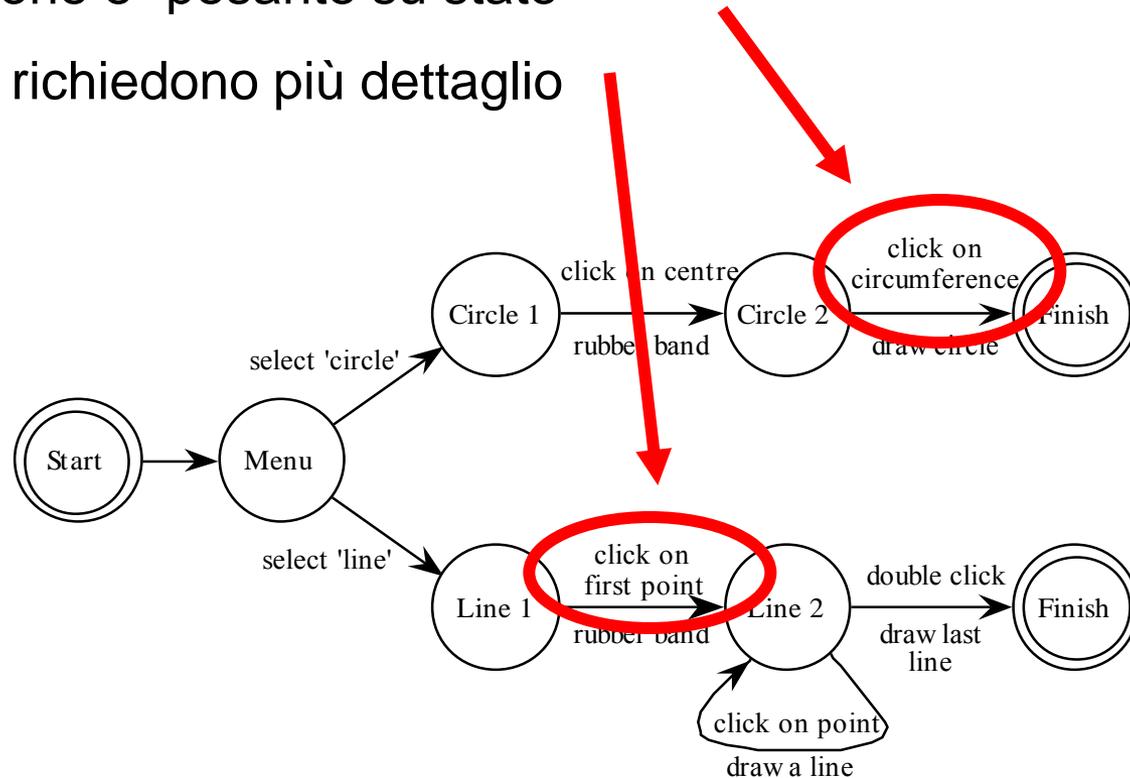
Reti di transizioni di stato (STN)

- Cerchi – stati
- Archi - azioni/eventi



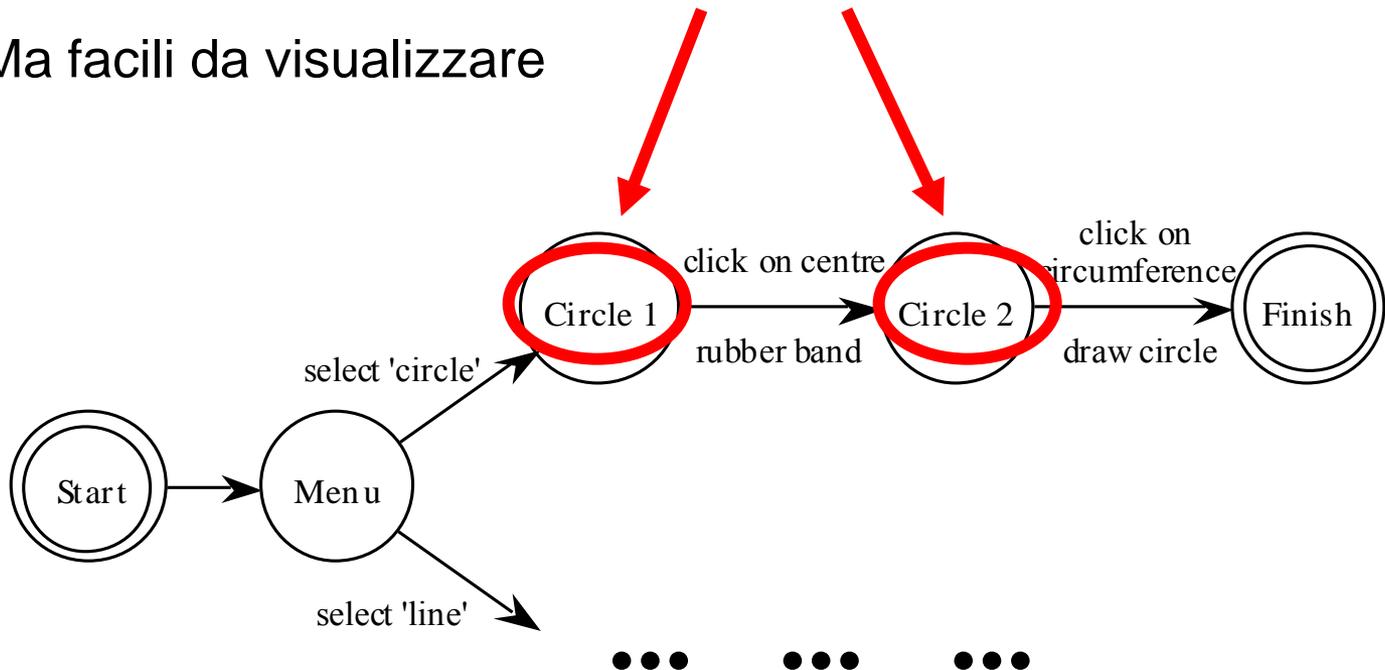
Reti di transizioni di stato - eventi

- Etichette su archi un po' compresse:
 - Notazione è “pesante su stato”
 - Eventi richiedono più dettaglio



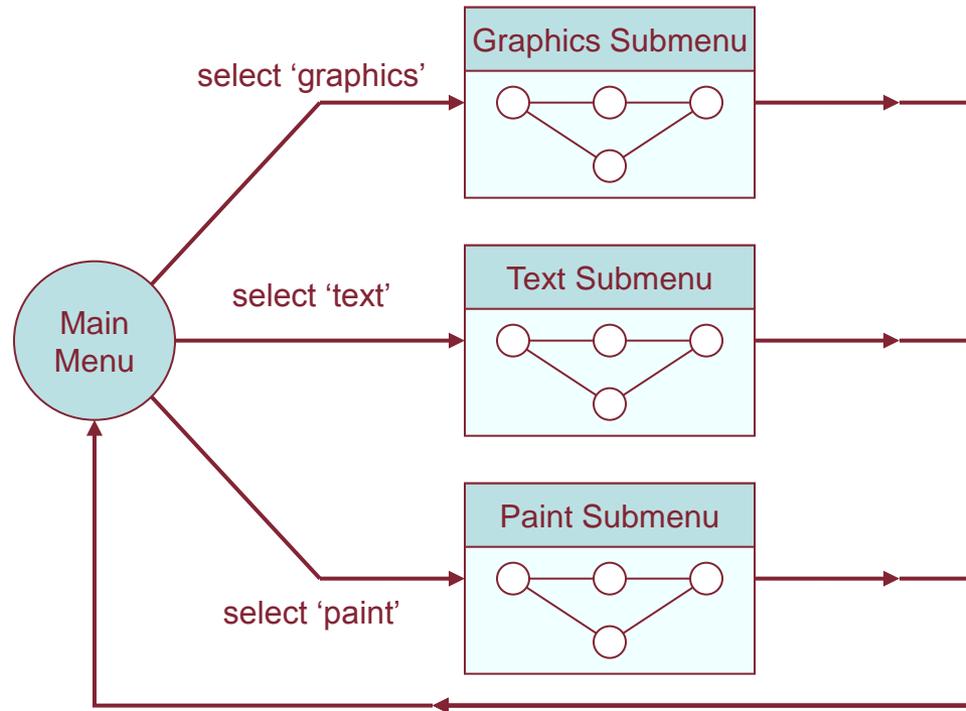
Reti di transizioni di stato - stati

- Etichette su stati poco informative:
 - Stati difficili da nominare
 - Ma facili da visualizzare



STN gerarchiche

- Gestione di dialoghi complessi
- Sotto-dialoghi identificati e nominati



Dialoghi concorrenti - I

semplice scatola di dialogo

Text Style

bold

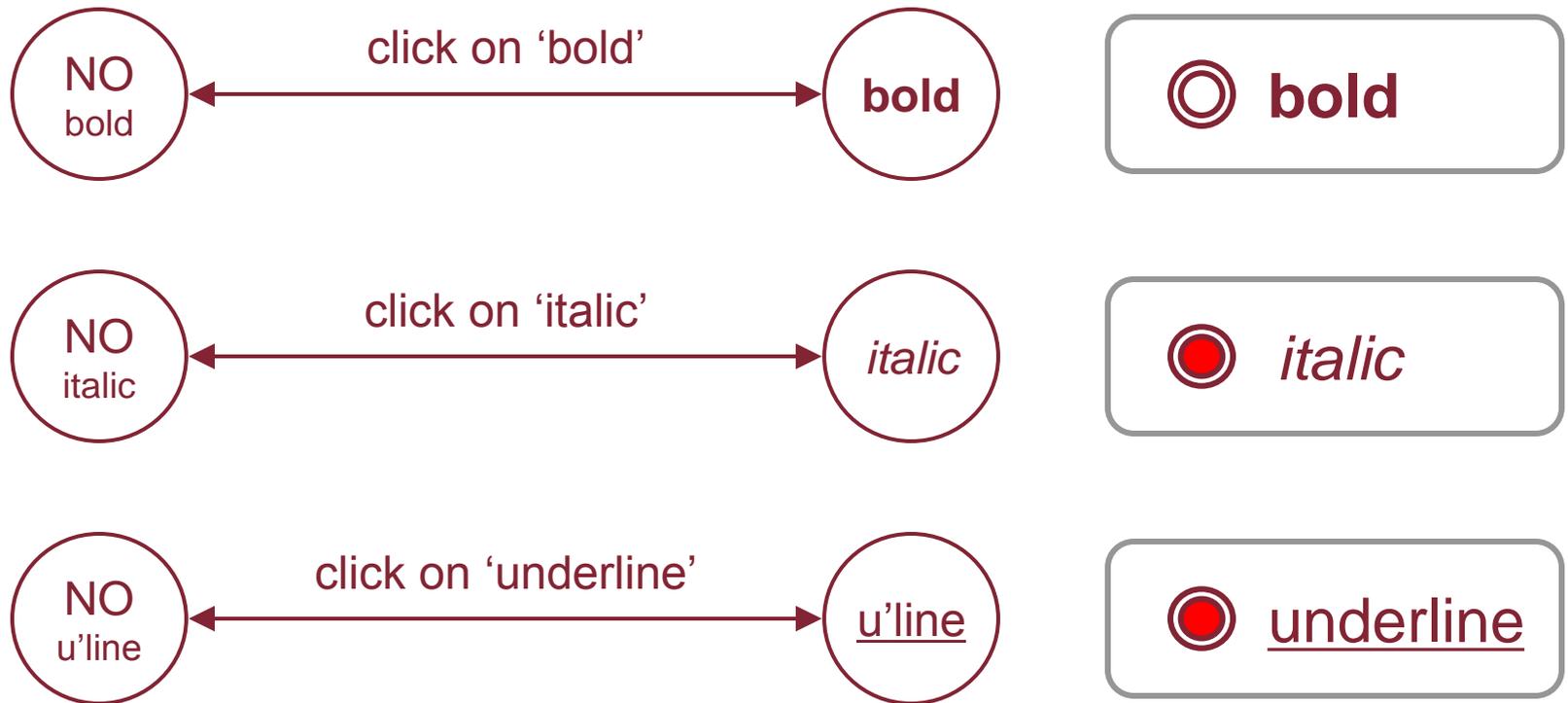
italic

underline

example

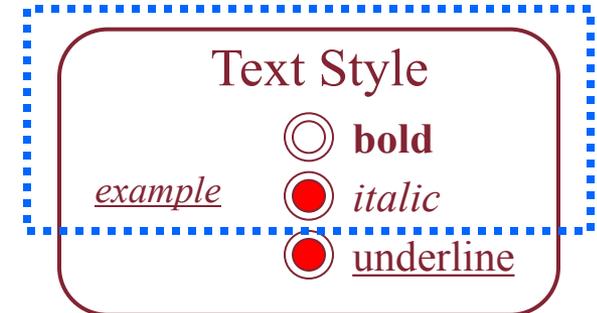
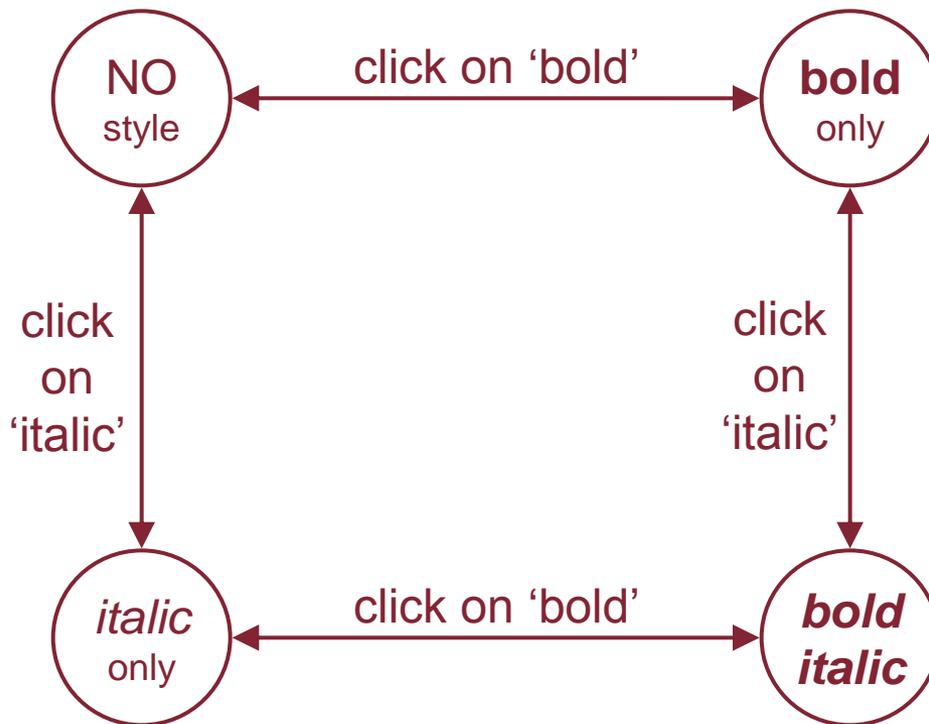
Dialoghi concorrenti – II

tre interruttori - STN individuali



Dialoghi concorrenti – III

grassetto e corsivo combinati



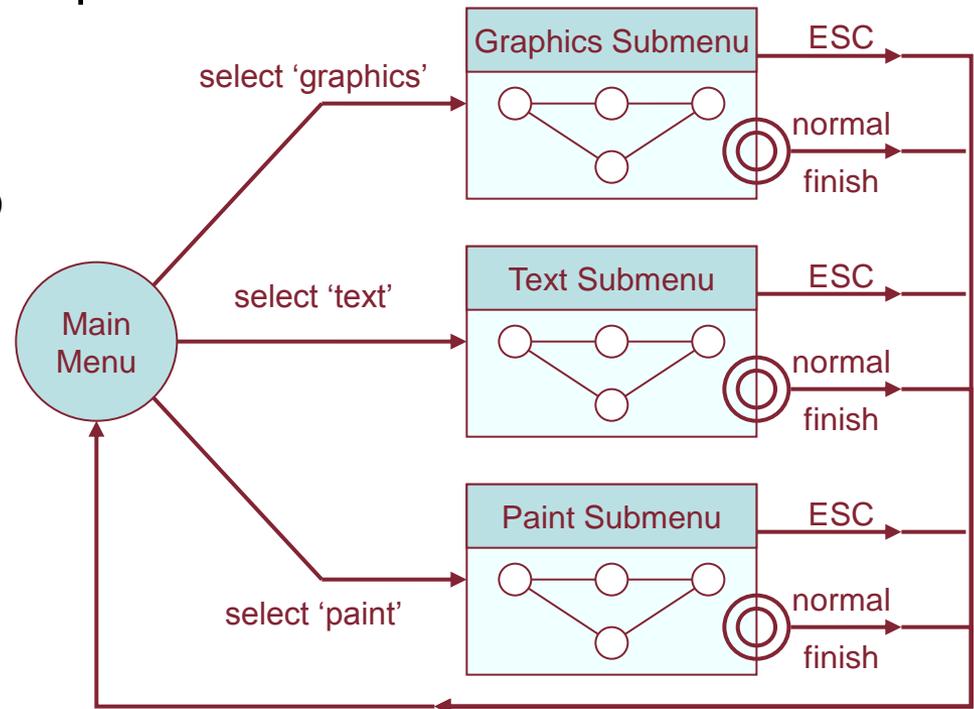
Fughe

- ‘back’ su Web, chiavi di fuga o cancellazione
 - Comportamento simile ovunque
 - Si finisce con spaghetti di comportamenti identici
- Per evitare

es. su diagramma ad alto livello

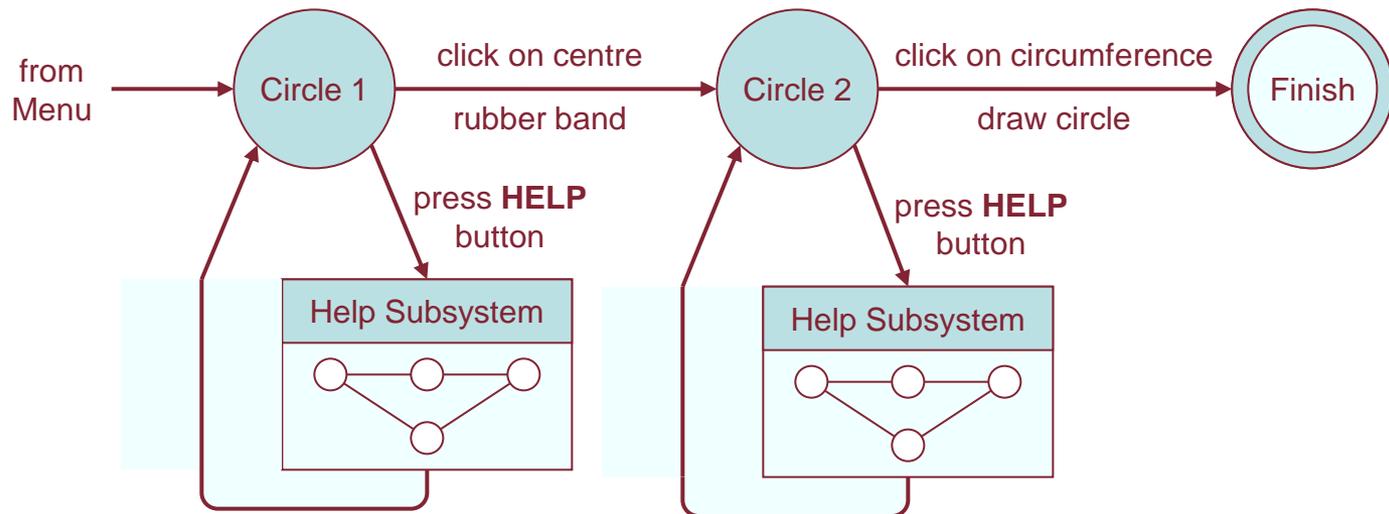
Uscita ‘normale’ per ogni menu

Più arco di fuga separato
attivo ‘ovunque’ nei sottomenu



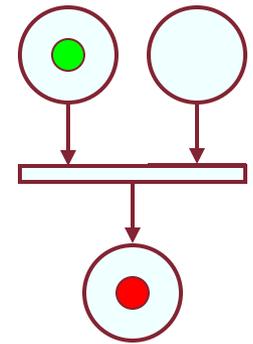
Menu di aiuto

- Problemi simili
 - Quasi stesso ovunque
 - Ma deve ritornare a punto diverso in dialogo
 - Si potrebbe specificare su STN ... ma crea disordine
 - Solitamente meglio aggiunto a livello “meta”

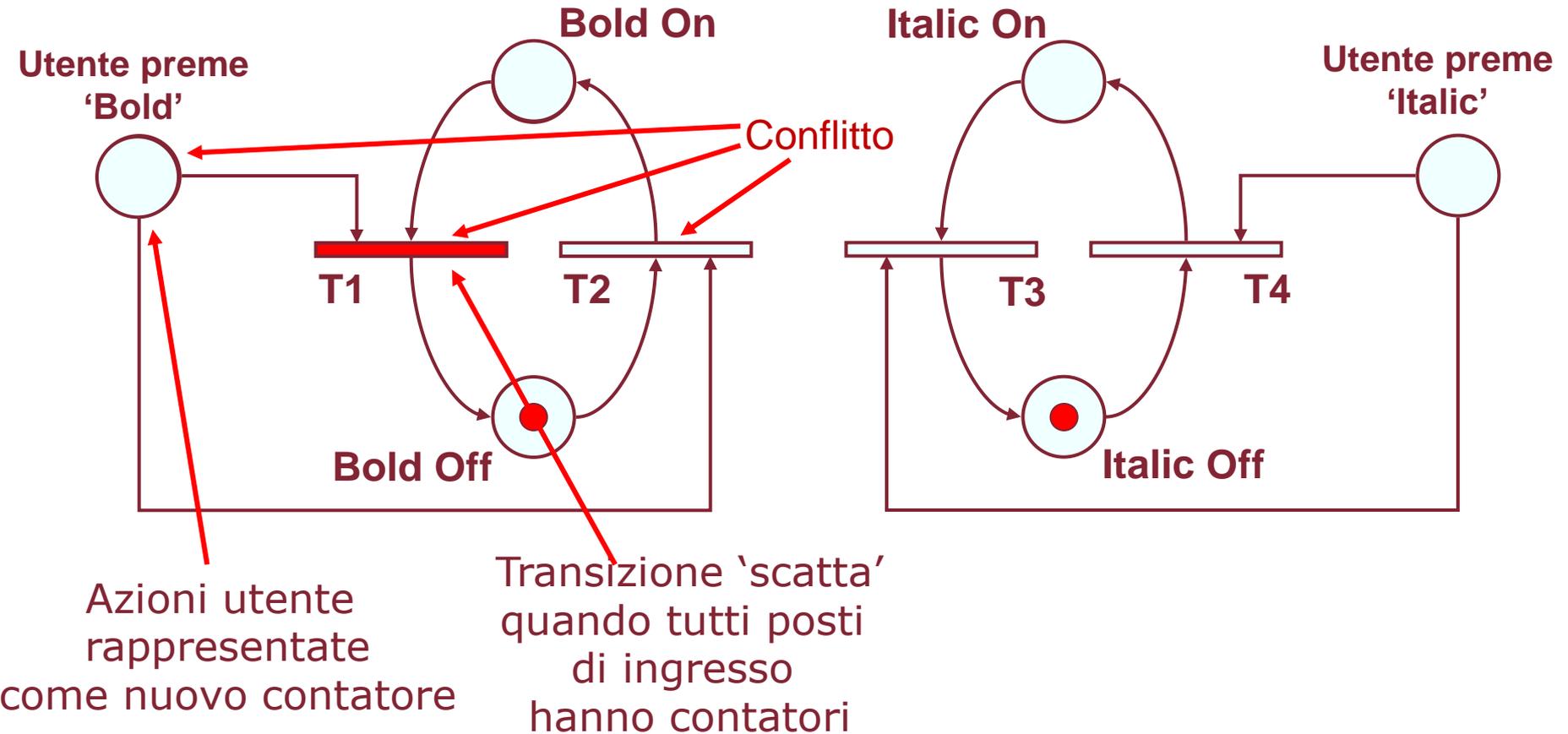


Reti di Petri

- Fra più vecchie notazioni in computazione
- Grafi di flusso:
 - Posti
 - Transizioni
 - Contatori
- simili a stati STN
- simili a archi STN
- su posti (stato corrente)
- Possibili contatori diversi
 - Stati di dialogo concorrenti
- Usati per specifica UI
 - Supporto a strumenti

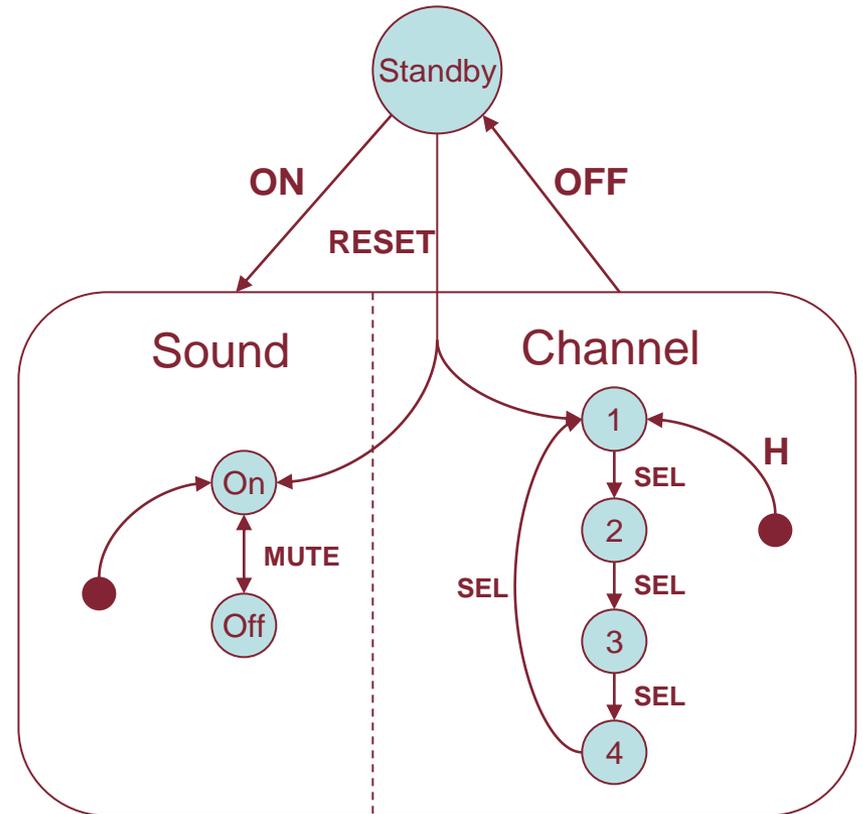


Esempio di rete di Petri



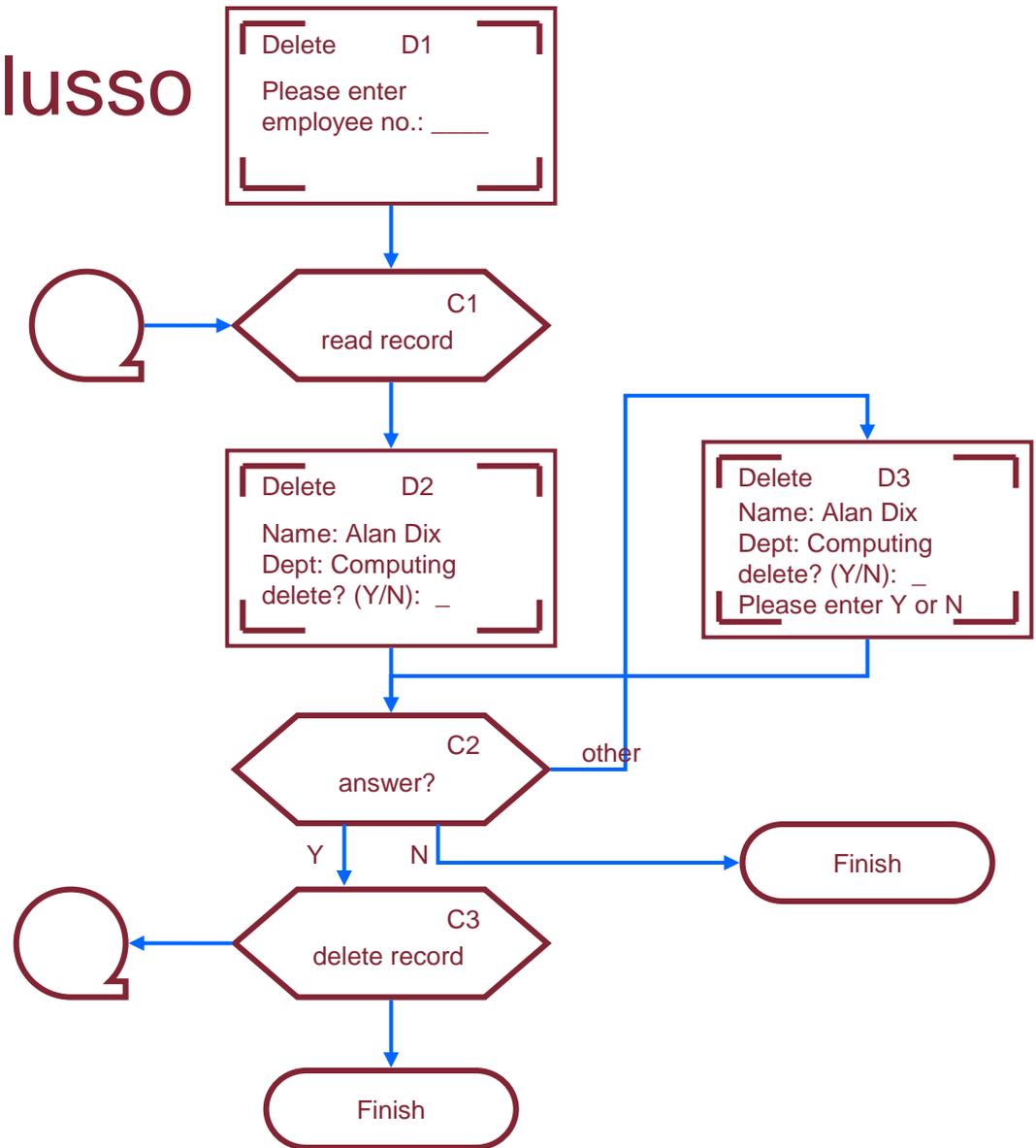
State charts

- Usate in UML
- Estensione a STN
 - Gerarchie
 - Sotto-reti concorrenti
 - Fughe
 - OFF sempre attivo
 - Storia
 - Collegamento marcato H
 - Va a ultimo stato quando si rientra in sottodialogo



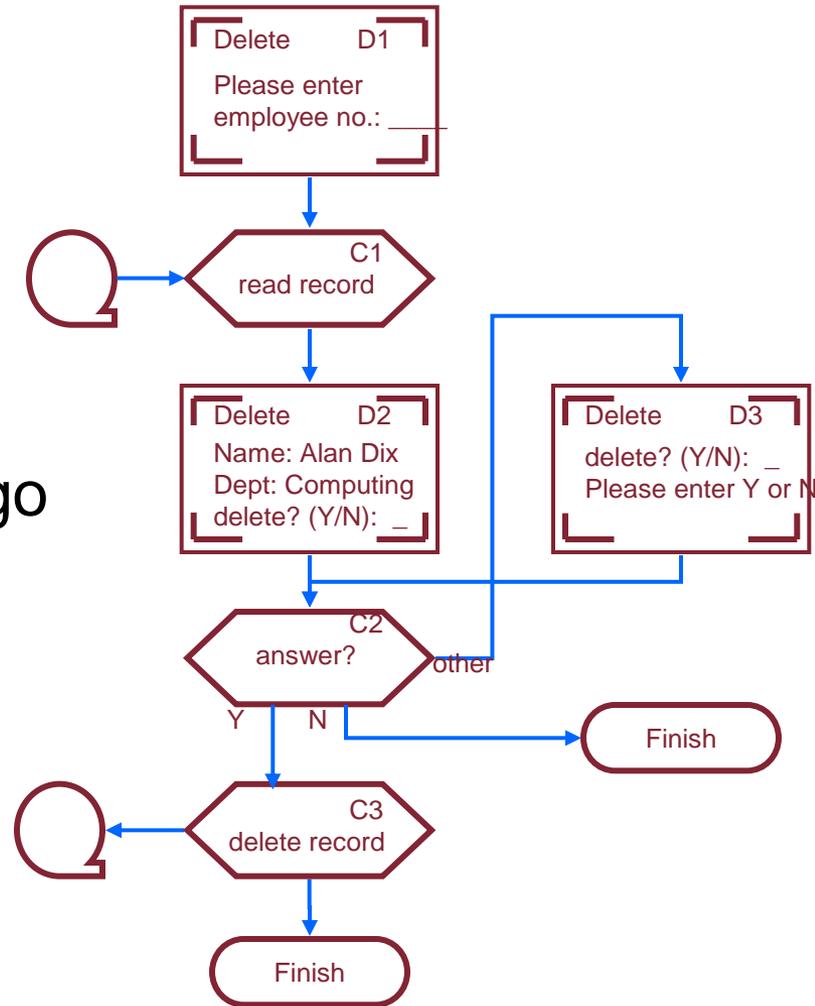
Diagrammi di flusso

- Familiari a programmatori
- Scatole
 - Processi/eventi
 - Non stati
- Uso per dialogo (non algoritmi interni)



Funziona!

- Notazioni formali – troppo lavoro?
- Elaborazione transizioni COBOL
 - Guidate da eventi – come interfacce
 - Struttura programma
≠ struttura dialogo
- Usa diagrammi di flusso del dialogo
 - Discutere con clienti
 - Trasformazione codice
 - Test sistematici
 - Guadagno di produttività 1000%
- Formalismo salva tempo!!



Notazioni testuali

Grammatiche

Regole di produzione

CSP e algebre di eventi

Grammatiche testuali

- Espressioni regolari

```
sel-line click click* dble-click
```

- BNF

```
mulSelect ::= select  
            | select mulSelect
```

- Più potenti di espressioni regolari o STN
- Ma SENZA dialogo concorrente
- Linguaggio eventi basso livello per azioni alto livello

Regole di produzione

- Insieme di regole non ordinate:

if condition then action

- Condizione basata su stato o eventi da servire
 - Ogni regola sempre potenzialmente attiva
- Buono per concorrenza
 - Cattivo per sequenza

Regole di produzione basate su eventi

Sel-line → first

C-point first → rest

C-point rest → rest

D-point rest → < draw line >

- Nota:
 - eventi aggiunti a lista di eventi pendenti
 - 'first' e 'rest' sono eventi generati internamente
- Cattivo per descrivere stato!

Sistema di produzione proposizionale

- Basato su stato
- Attributi:
 - Mouse: { mouse-off, select-line, click-point, double-click }
 - Line-state: { menu, first, rest }
- Regole (senza indicazione risposta):
 - select-line → mouse-off first
 - click-point first → mouse-off rest
 - click-point rest → mouse-off
 - double-click rest → mouse-off menu
- Cattivo per eventi!

CSP e algebre di processo

- Buono per dialoghi sequenziali

```
Bold-tog = select-bold? → bold-on → select-bold? →  
bold-off → Bold-  
tog
```

```
Italic-tog = . . .
```

```
Under-tog = . . .
```

- e dialogo concorrente

```
Dialogue-box = Bold-tog || Italic-tog || Under-tog
```

- Ma causalità non chiara

Notazioni per dialogo - Sommario

- Diagrammatiche
 - STN, Flow charts
- Testuali
 - grammatiche, regole di produzione, CSP
- Problemi
 - Basate su eventi vs. basate su stato
 - Potenza vs. chiarezza
 - Modello vs. notazione
 - Sequenziale vs. concorrente

Semantica - Alexander SPI (Specifying and Prototyping Interaction) (i)

- Specifica in due parti:
 - EventCSP – puro ordine del dialogo
 - EventISL – semantica dipendente da bersaglio
(Interaction specification language)
- Descrizione dialogo – centralizzata
- Compromesso sintassi/semantica - tollerabile

Semantica - Alexander SPI (ii)

- **EventCSP**

```
Login = login-mess -> get-name -> Passwd
```

```
Passwd = passwd-mess -> (invalid -> Login [] valid -> Session)
```

- **EventISL**

```
event: login-mess
```

```
  prompt: true
```

```
  out: "Login:"
```

```
event: get-name
```

```
  uses: input
```

```
  set: user-id = input
```

```
event: valid
```

```
  uses: input, user-id, passwd-db
```

```
  wgen: passwd-id = passwd-db(user-id)
```

```
event: invalid
```

```
  uses: input, user-id, passwd-db
```

```
  wgen: passwd-id != passwd-db(user-id)
```

Semantica – codice puro

- Ciclo di eventi per elaboratore testi
- Descrizione dialogo
 - molto distribuita
- Compromesso sintassi/semantica
 - terribile!

```
switch ( ev.type ) {
  case button_down:
    if ( in_text ( ev.pos ) ) {
      mode = selecting;
      mark_selection_start(ev.pos);
    }
    ...
  case button_up:
    if ( in_text ( ev.pos )
        && mode == selecting ) {
      mode = normal;
      mark_selection_end(ev.pos);
    }
    ...
  case mouse_move:
    if (mode == selecting ) {
      extend_selection(ev.pos);
    }
    ...
} /* end of switch */
```

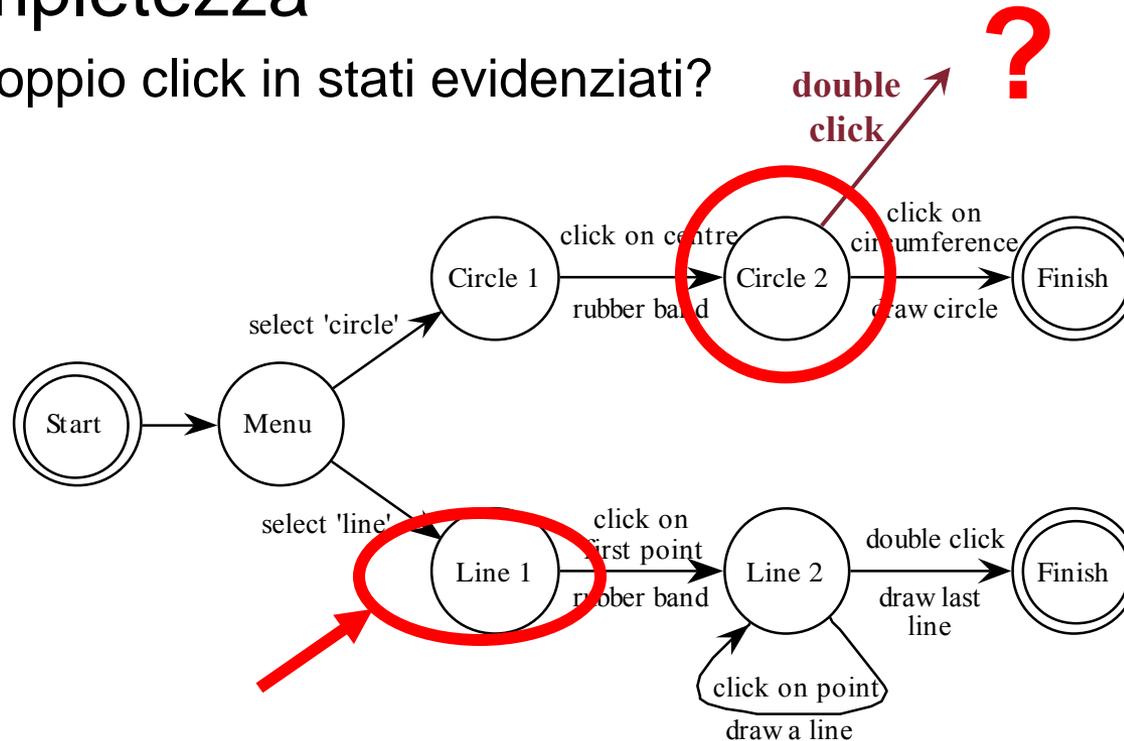
Proprietà delle azioni

- Completezza
 - Archi mancanti
 - Circostanze impreviste
- Determinismo
 - Più archi per azione
 - Deliberato: decisione di applicazioni
 - Casualità: regole di produzione
- Fughe annidate
- Coerenza
 - Stessa azione, stesso effetto?
 - Modi e visibilità

Controllo delle proprietà (i)

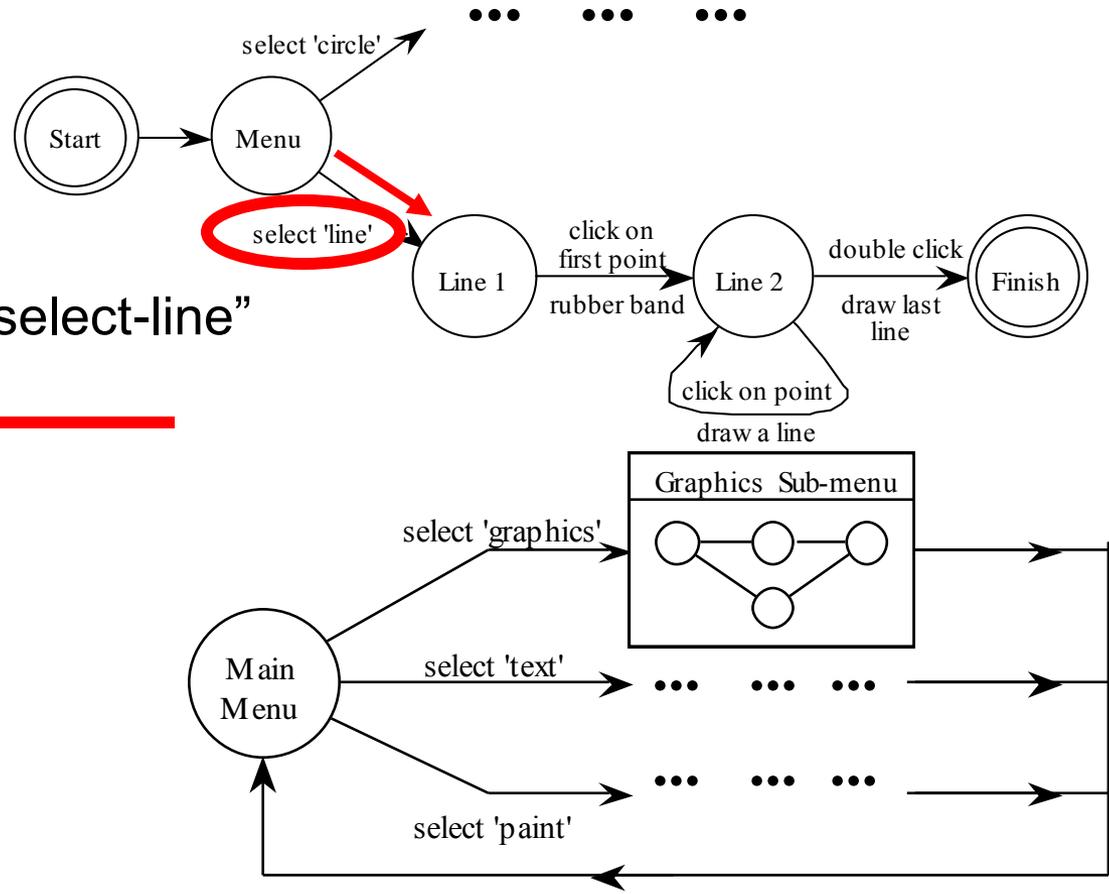
- Completezza

- Doppio click in stati evidenziati?



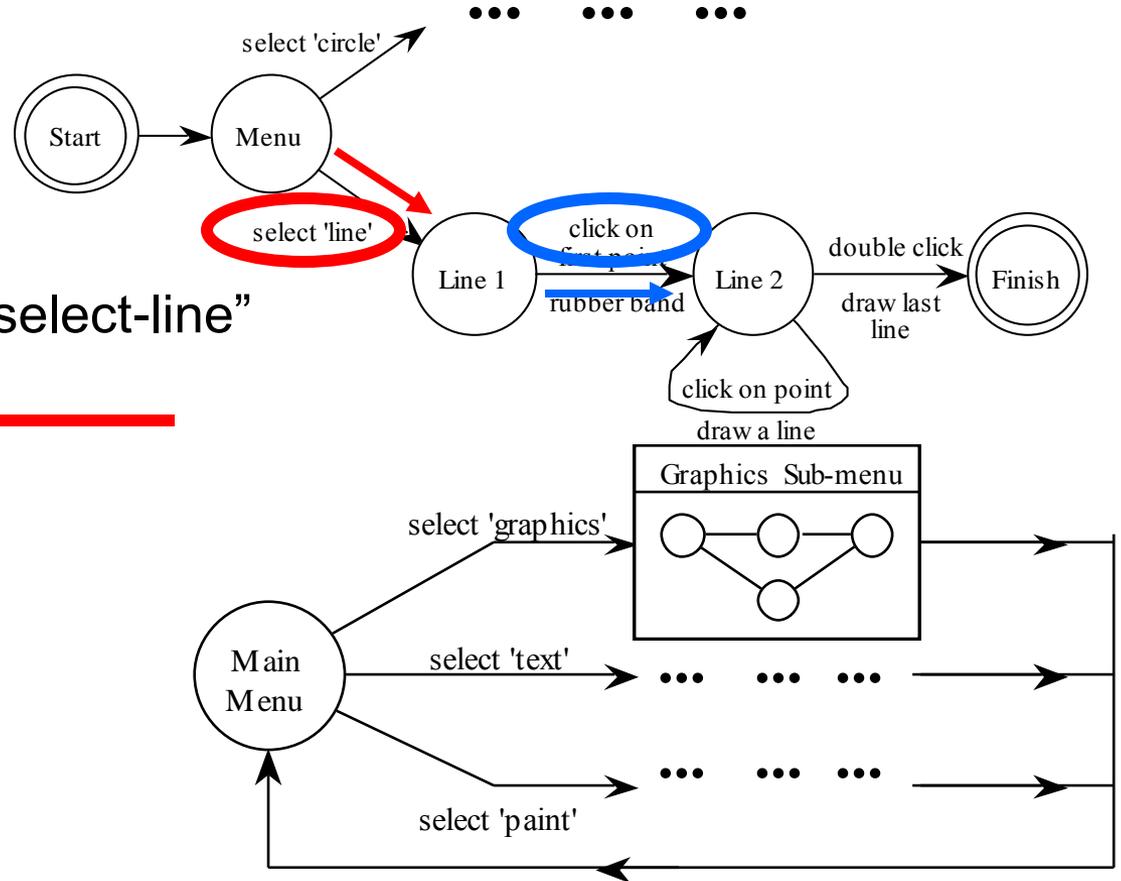
Controllo delle proprietà (ii)

- Reversibilità:
 - Per invertire “select-line”



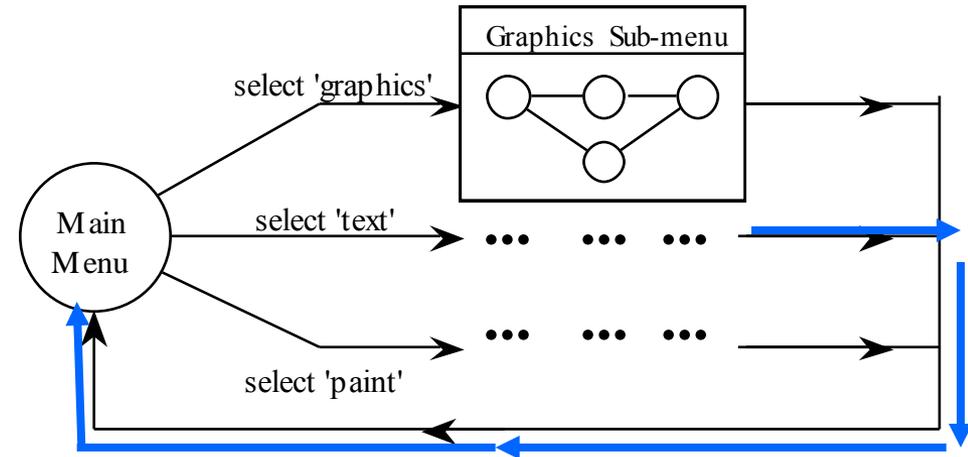
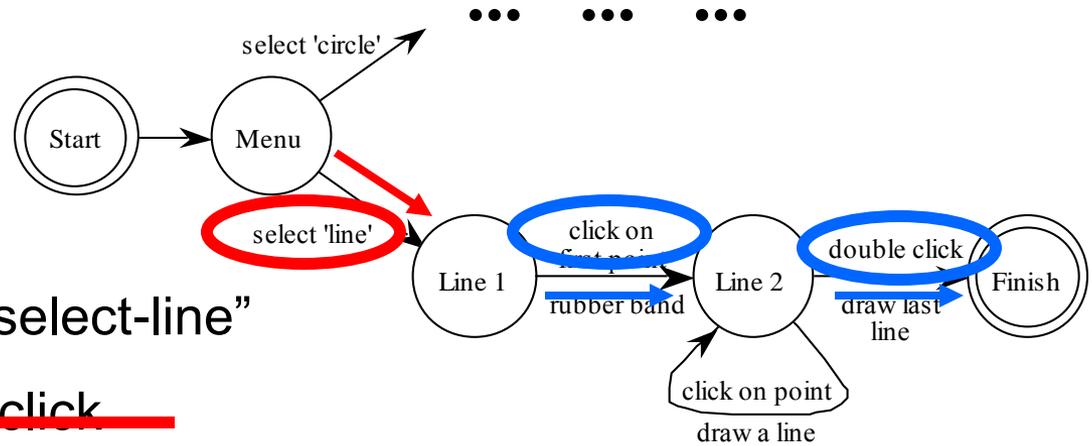
Controllo delle proprietà (ii)

- Reversibilità:
 - Per invertire “select-line”
 - click



Controllo delle proprietà(ii)

- Reversibilità:
 - Per invertire “select-line”
 - ~~click - doppio click~~

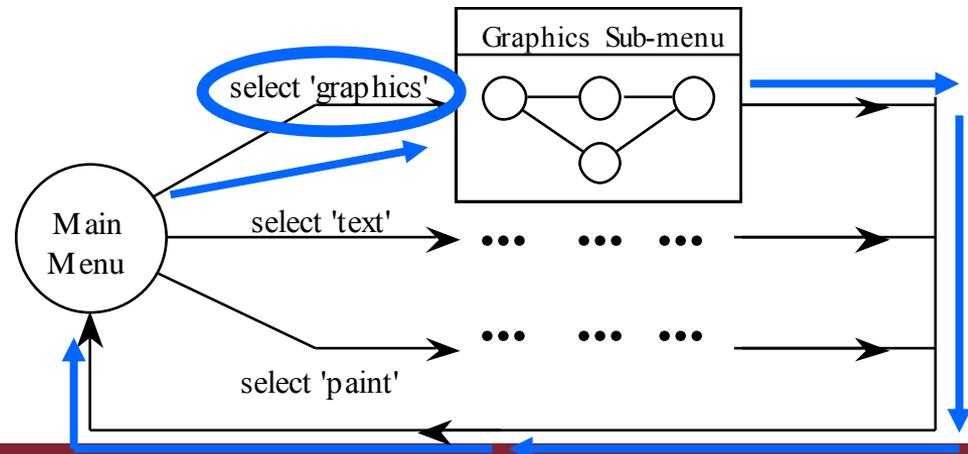
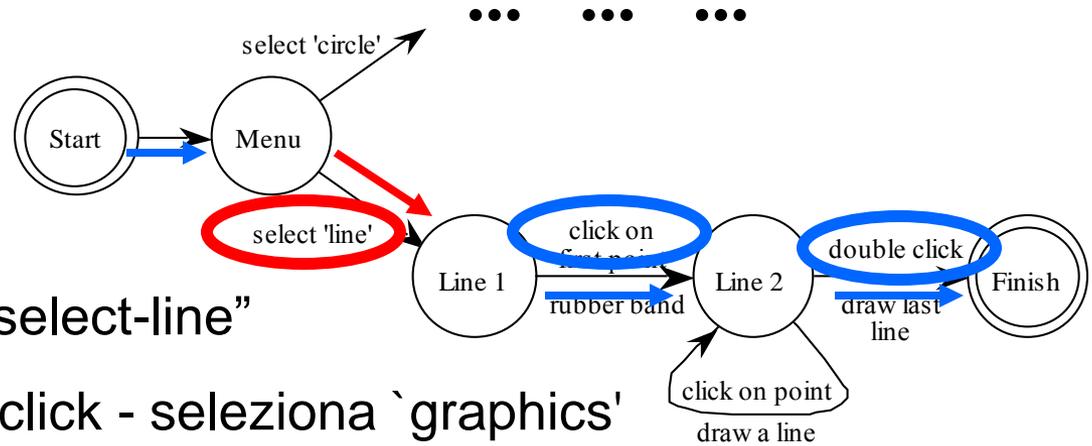


Controllo delle proprietà (ii)

- Reversibilità:

- Per invertire “select-line”
- click - doppio click - seleziona `graphics`
- (3 actions)

- N.B. non undo

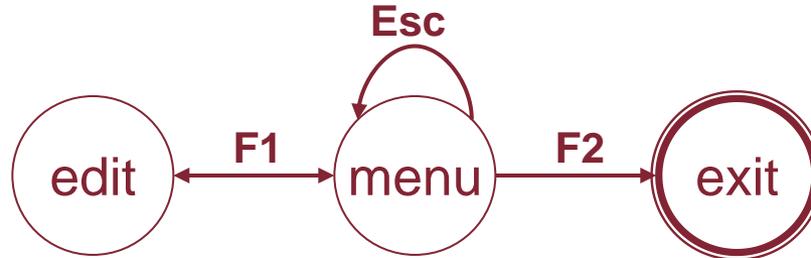


Proprietà degli stati

- Raggiungibilità
 - Si può raggiungere qualsiasi punto da qualsiasi punto?
 - E quanto facilmente?
- Reversibilità
 - Si può raggiungere stato precedente?
 - Ma NON con UNDO?
- Stati pericolosi
 - Stati in cui non si vuole arrivare

Stati pericolosi

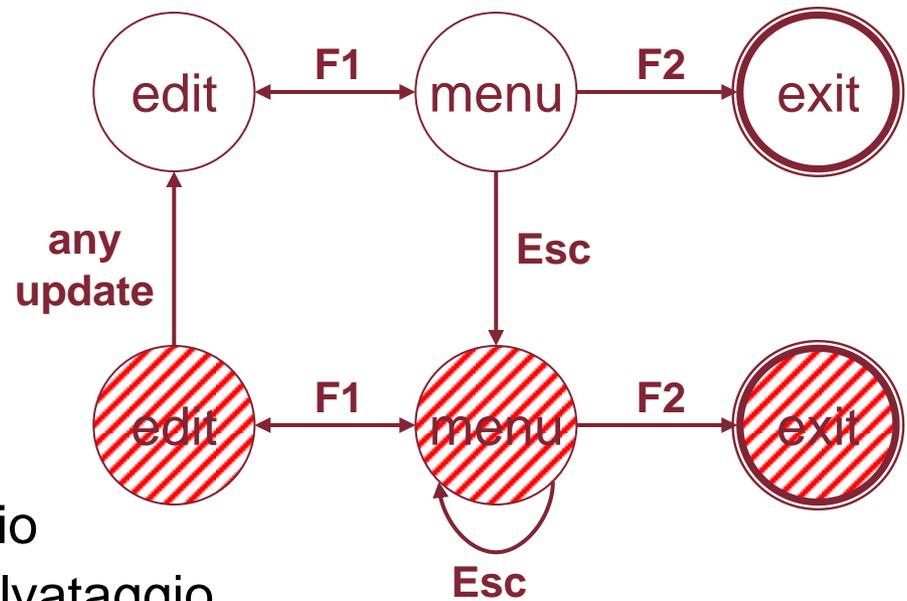
- Elaboratori di testi: due modi e uscita
 - F1 - cambia modo
 - F2 - esce (e salva)
 - Esc - nessun cambiamento di modo



ma ... Esc riazzera autosave

Stati pericolosi (ii)

- Esce con/senza salvataggio \Rightarrow stati pericolosi
- Stati duplicati – distinzione semantica



F1-F2 - esce con salvataggio

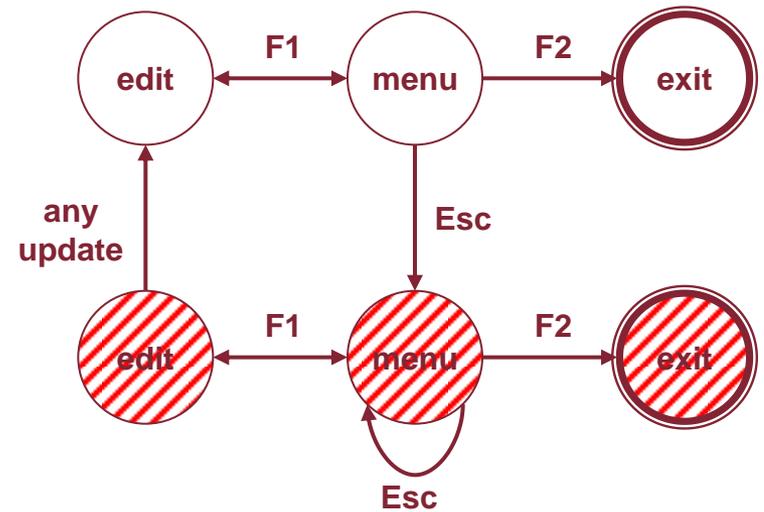
F1-Esc-F2 – esce senza salvataggio

Aspetti lessicali

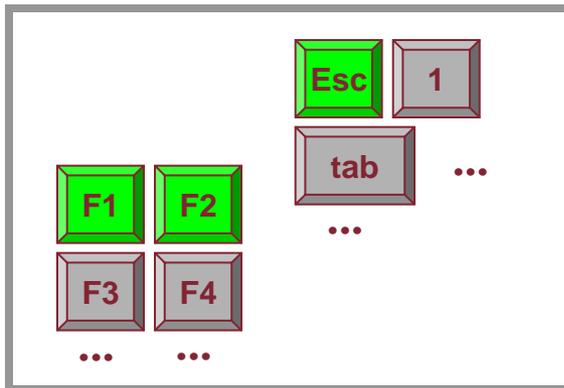
- **Visibilità**
 - Differenziare modi e stati
 - Annotazioni a dialogo
- **Stile**
 - Comando – verbo nome
 - Basata su mouse – nome verbo
- **Disposizione**
 - Non solo apparenza ...

Aspetti di disposizione

- Elaboratore di testi – stati pericolosi

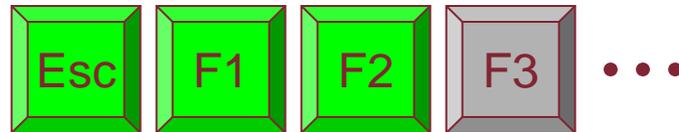


- Vecchia tastiera - OK



Aspetti di disposizione

- Nuova disposizione di tastiera



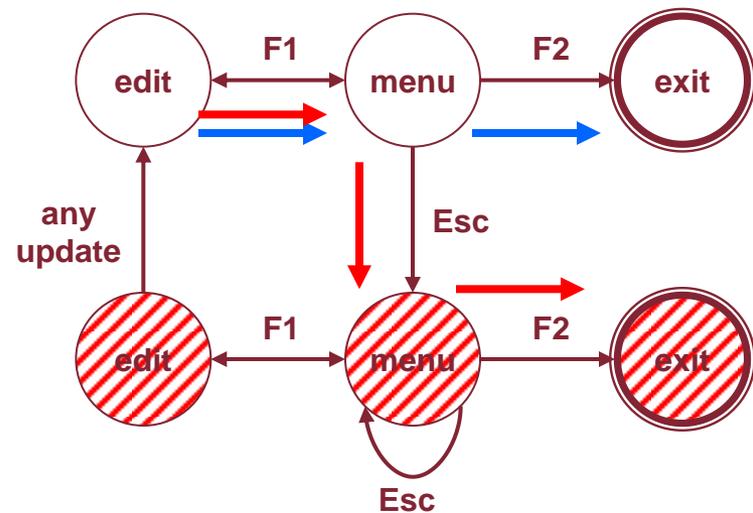
Intende F1-F2 (sicuro)



Dito incontra Esc



F1-Esc-F2 – disastro!

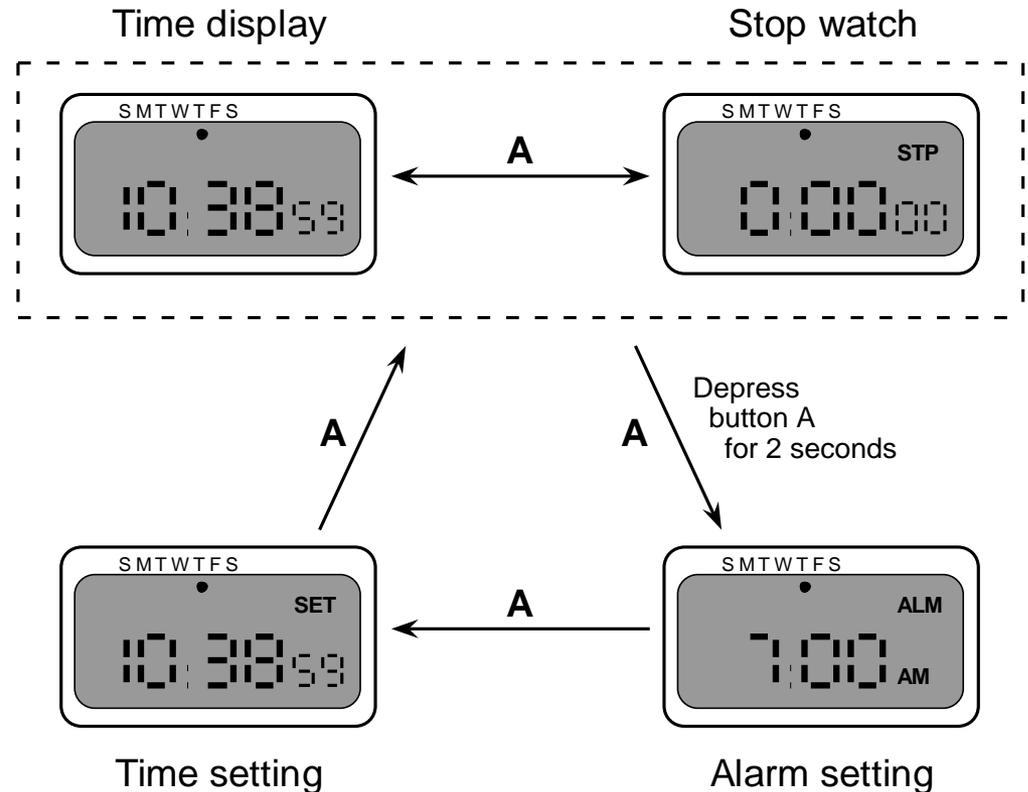


Analisi del dialogo – Sommario

- **Semantica e dialogo**
 - Attaccare semantica
 - Descrizione dialogo distribuita / centralizzata
 - Massimizzare descrizione sintattica
- **Proprietà del dialogo**
 - Proprietà azioni: completezza, determinismo, coerenza
 - Proprietà stato: raggiungibilità, reversibilità, stati pericolosi
- **Aspetti di presentazione e stile**
 - Visibilità, stile, disposizione
 - N.B. non indipendente da dialogo

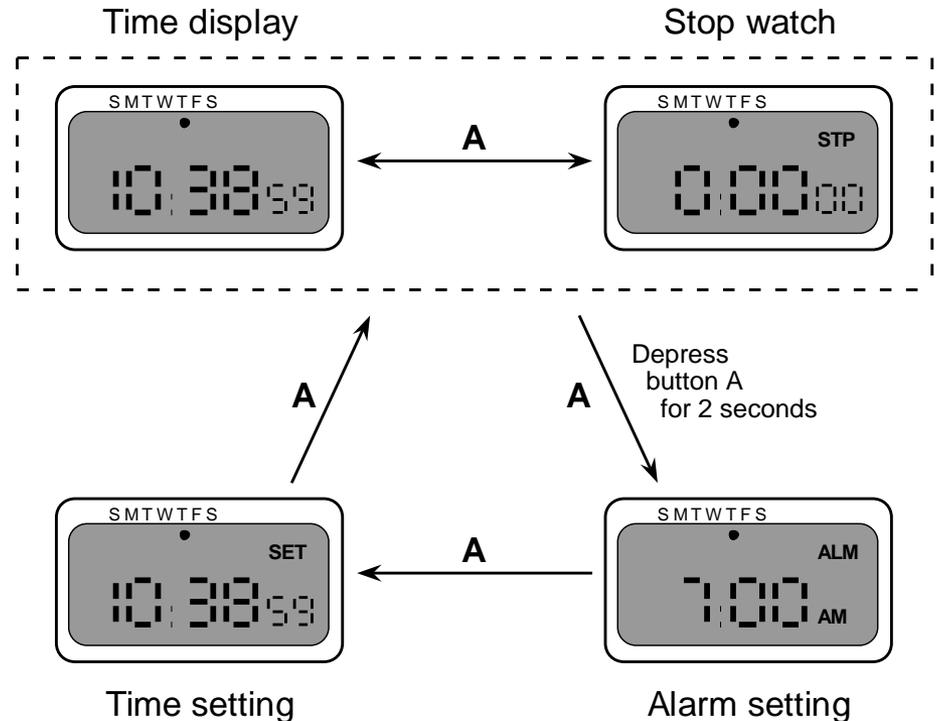
Orologio digitale – Istruzioni utente

- Due modalità
- Interfaccia limitata - 3 pulsanti
- Pulsante A cambia modalità



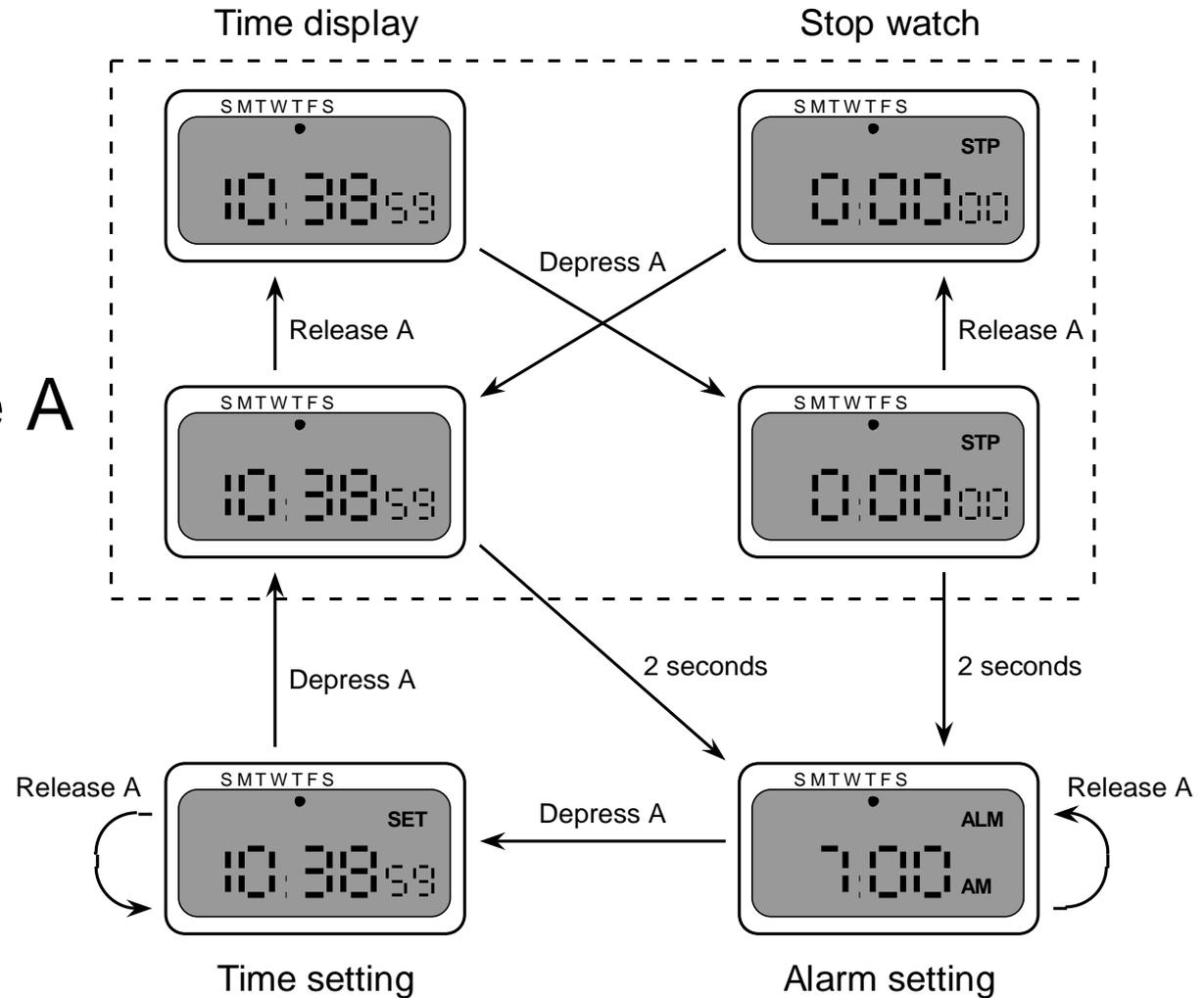
Orologio digitale – Istruzioni utente

- Stati pericolosi
 - *controllati*
... pressione di due secondi
- Completezza
 - Distingue premi A e rilascia A
 - Cosa fanno nei vari modi?



Orologio digitale – Istruzioni progettista

Per pulsante A



Fondamenti logici del progetto I

Informazione che spiega perché sistema fatto in certo modo.

Benefici fondamento logico

- Comunicazione attraverso ciclo di vita
- Riutilizzo di conoscenza di progetto fra prodotti
- Costringe a disciplina di progetto
- Presenta argomenti per compromessi di progetto
- Organizza spazio di progetto potenzialmente grande
- Cattura informazione di contesto

Fondamenti logici del progetto II

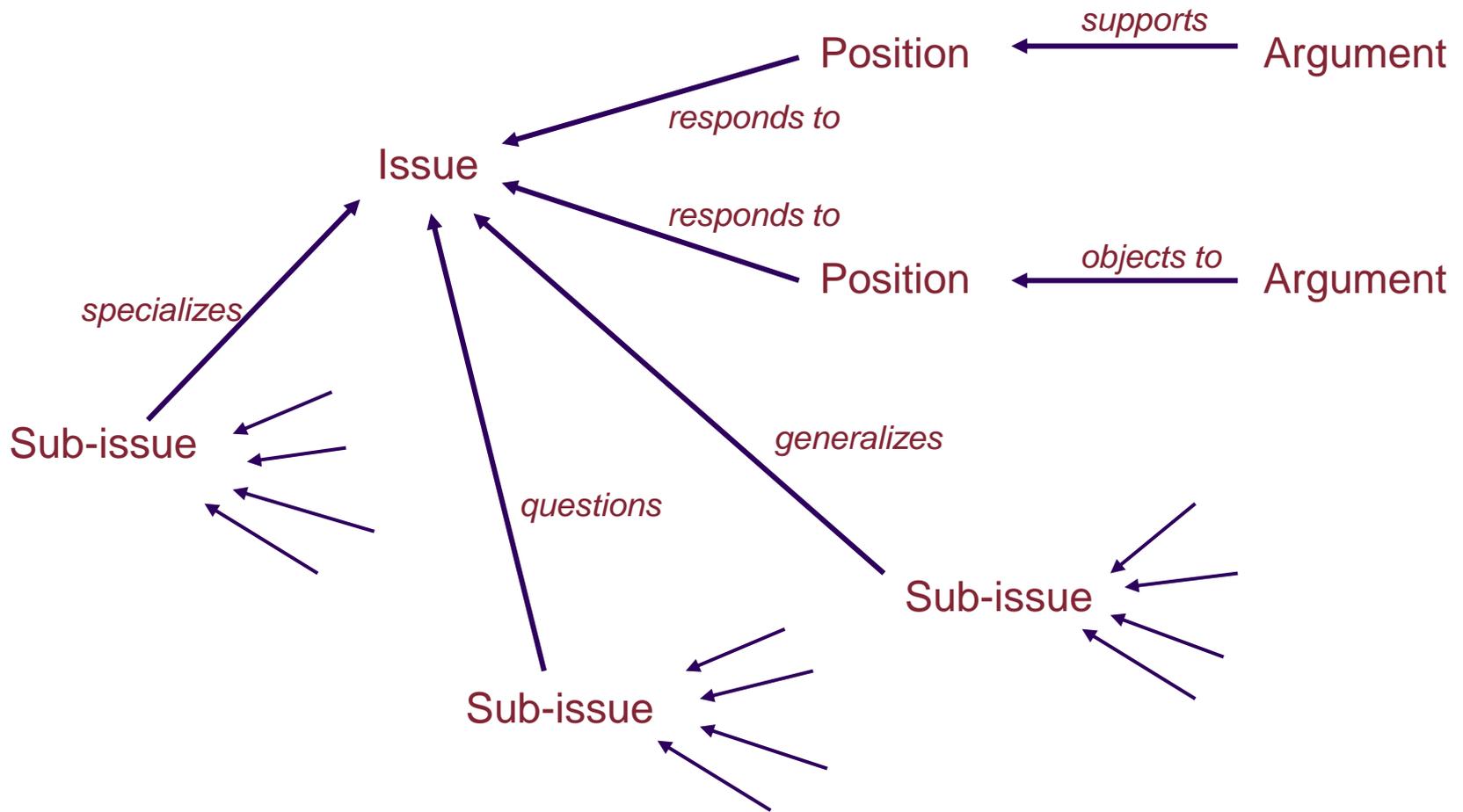
Tipi di fondamento

- Orientato a processo
 - Mantiene ordine di scelta e formazione decisioni
- Orientato a struttura
 - Enfattizza strutturazione a posteriori alternative considerate
- Due esempi:
 - Sistema informativo basato sui problemi (IBIS)
 - Analisi spazio di progetto

Issue-based information system (IBIS)

- Base per gran parte ricerca su fondamenti logici
- Orientato a processo
- Elementi principali:
 - Questioni
 - Struttura gerarchica con una questione "radice"
 - Posizioni
 - Soluzioni potenziali per questione
 - Argomenti
 - Modifica relazione tra posizioni e questioni
- gIBIS versione grafica

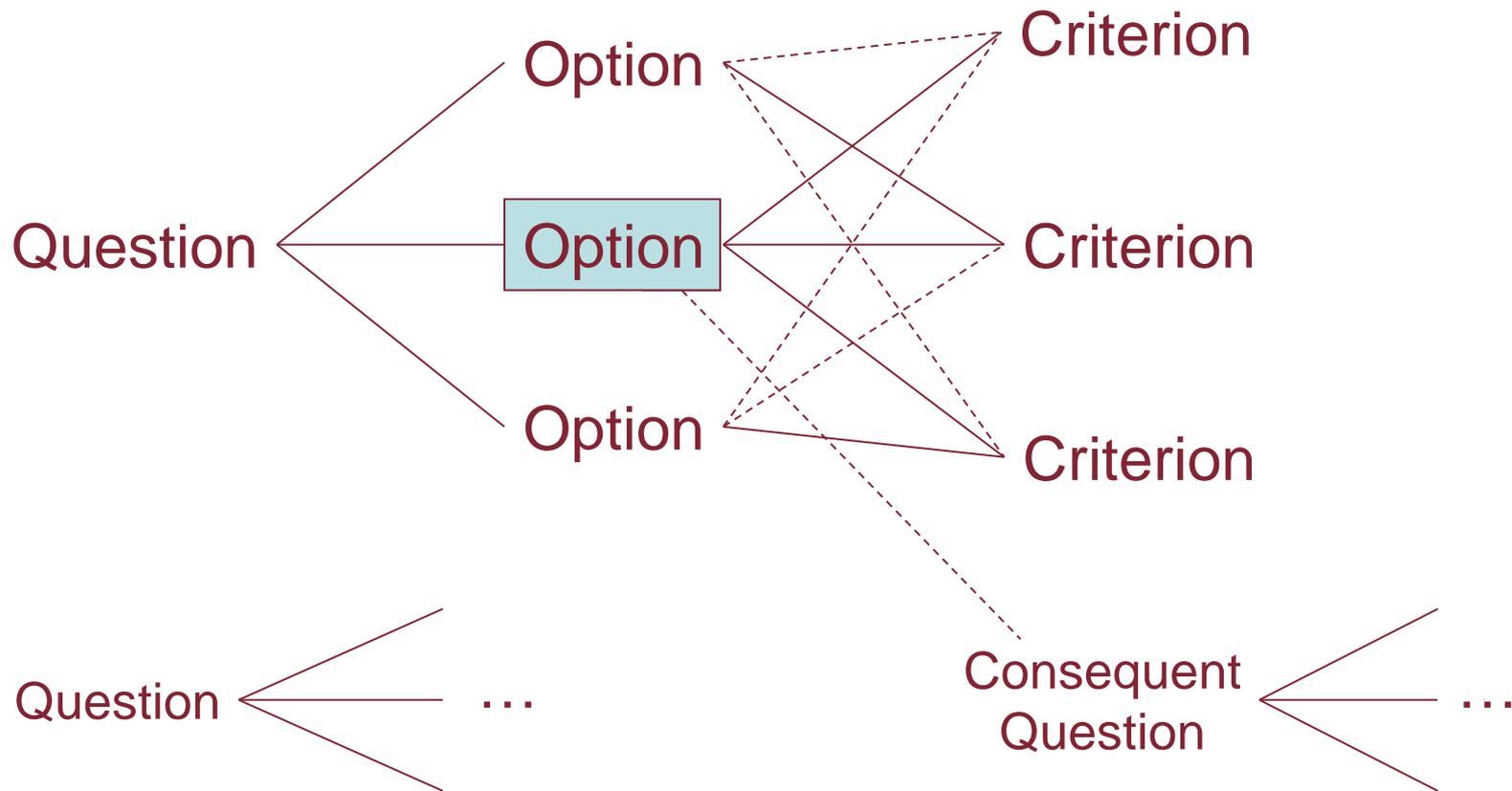
Struttura di gIBIS



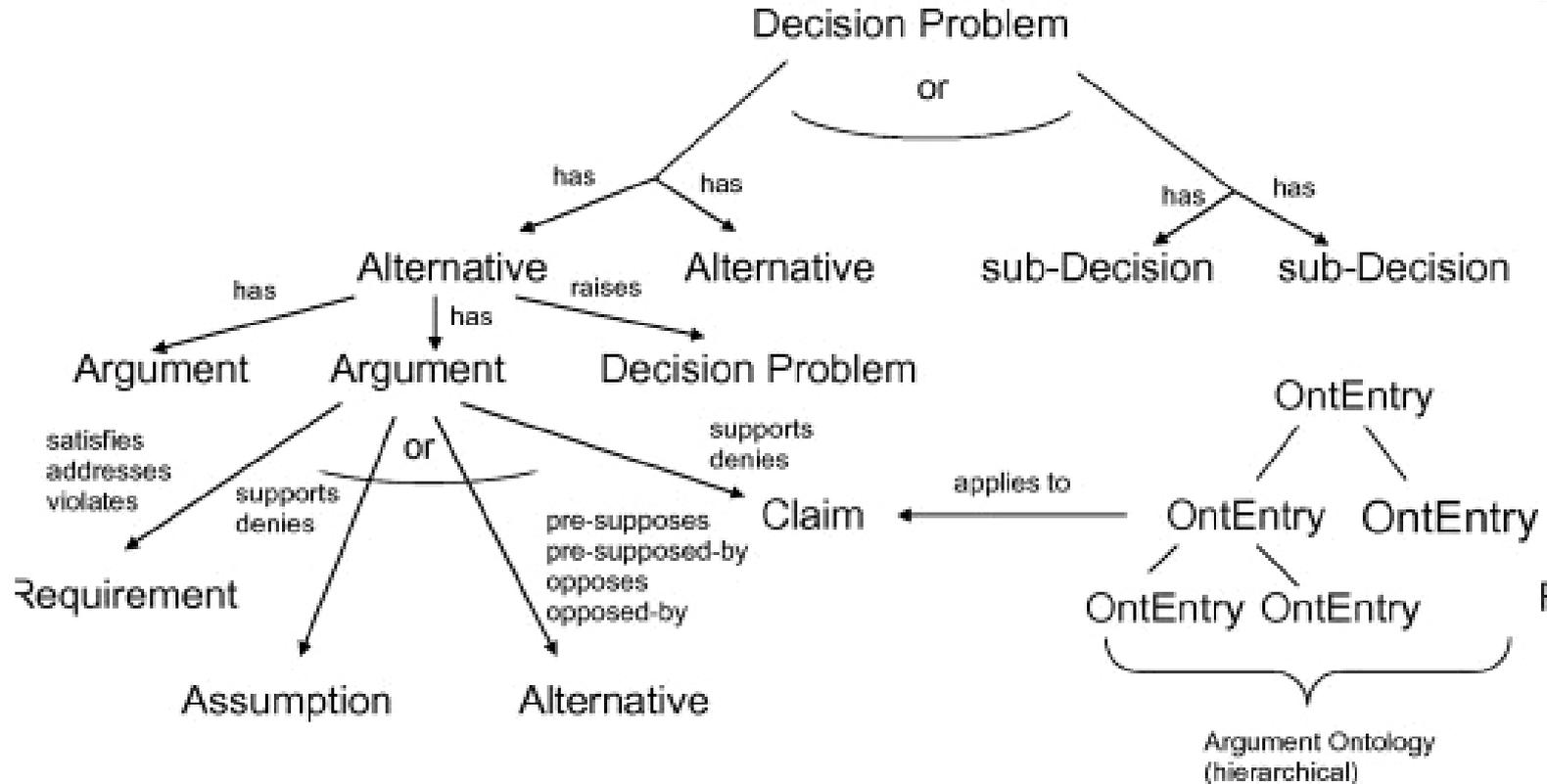
Analisi dello spazio di progetto

- Orientato a struttura
- QOC – struttura gerarchica:
 - Q: Domande (e sotto-domande)
 - rappresentano questioni principali di un progetto
 - O: Opzioni
 - forniscono soluzioni alternative alle domande
 - C: Criteri
 - mezzi per valutare le opzioni per fare una scelta
- DRL – simile a QOC con linguaggio più ampio e semantica più formalizzata
 - Decision representation language

Notazione QOC



Decision Representation Language



Da: Janet E. Burge, David C. Brown, Software Engineering Using RATIONALE, *Journal of Systems and Software*, **81**(3):395-413.