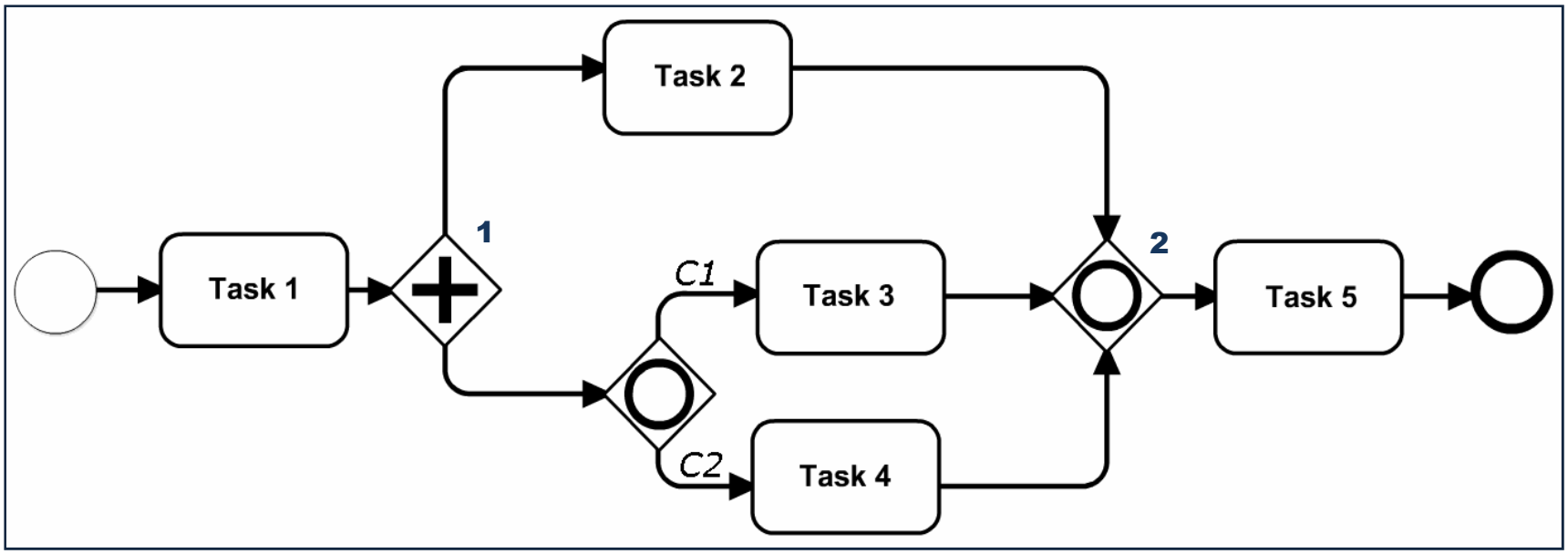
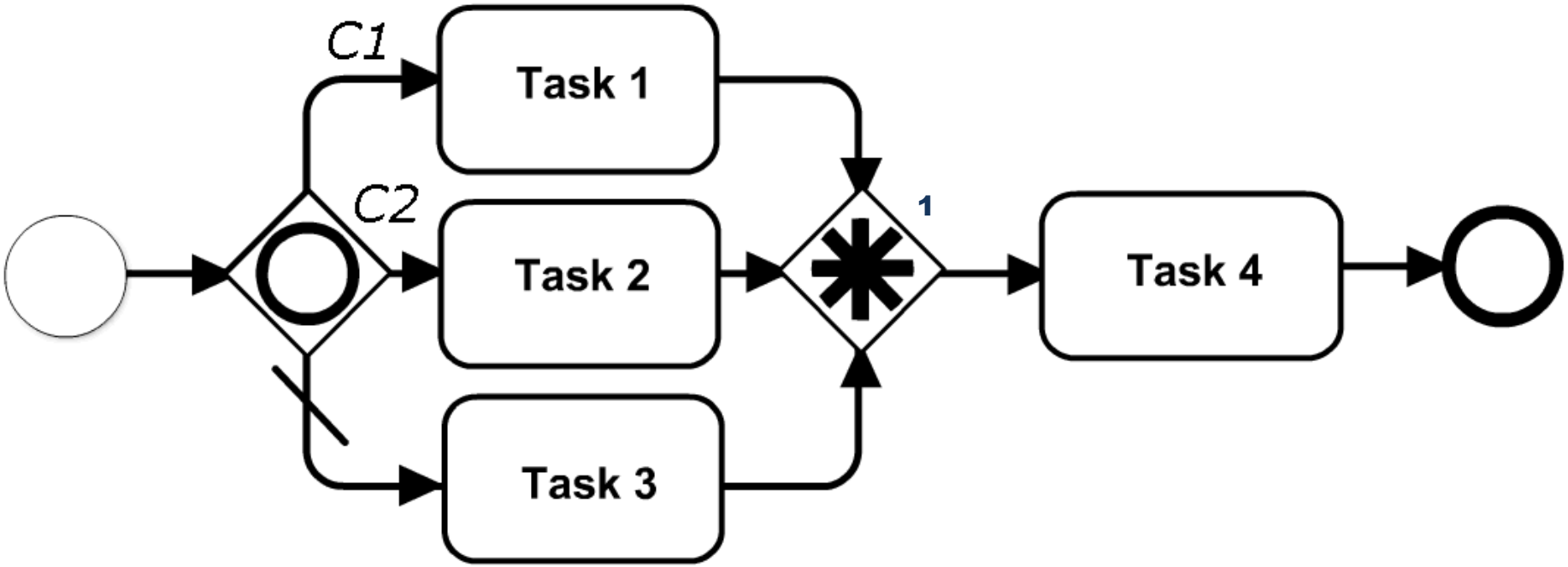
- Il *Parallel Gateway*[1] non ha condizioni logiche associate. Viene usato per sincronizzare flussi paralleli (in ingresso) e per creare flussi paralleli (in uscita). Si attende che tutti i token in ingresso siano arrivati, e poi si emettono nuovi token su tutti i flussi in uscita.
- Nell'esempio di seguito, si noti come il secondo gateway inclusivo[2] attenda tutti e soli i token prodotti prima di procedere. Se sostituiamo ad esso un gateway parallelo, nel caso di token non prodotto (*C1* o *C2* pari a false) il token in uscita non verrebbe mai emesso.



http://www.iet.unipi.it/m.cimino/ixl/res/mov09.swf

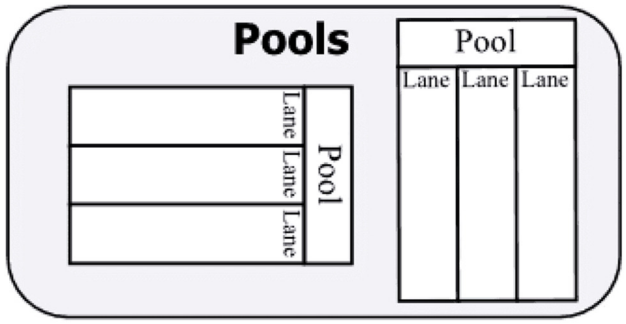
- Il *Complex Gateway* descrive casi decisionali complessi che richiederebbero la combinazione di molti gateway. Per evitare ciò, il comportamento del complex gateway può essere descritto usando il linguaggio naturale.



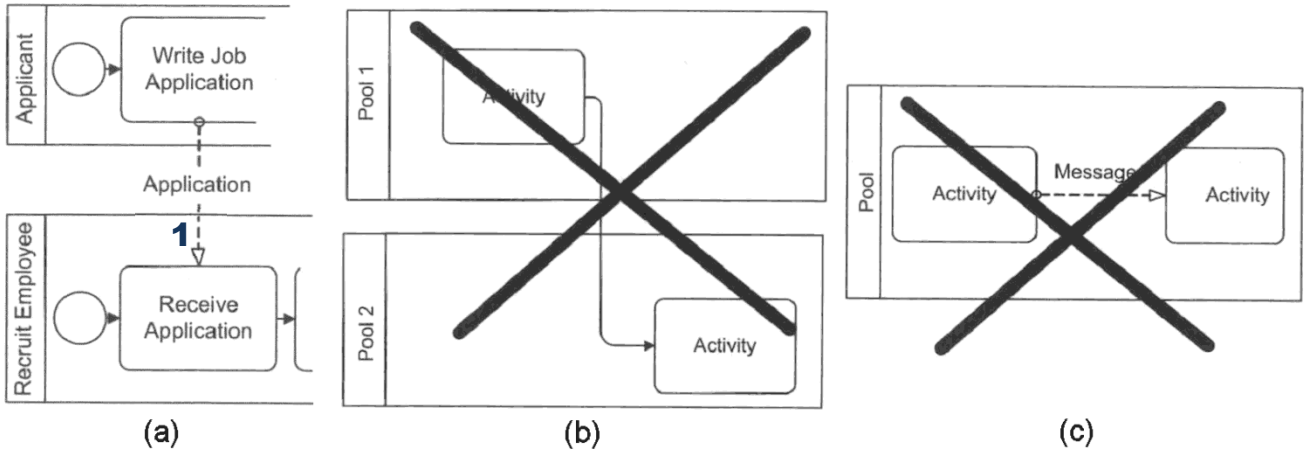
http://www.iet.unipi.it/m.cimino/bpm/res/mov10.swf

- Il complex gateway potrebbe essere usato per gestire qualsiasi situazione. Tuttavia è buona norma evitarlo poiché rende i modelli di processo meno leggibili.

- Le *Swimlane* sono usate per organizzare le attività e rappresentare collaborazioni tra partner. Un processo è organizzato dentro un *pool* di swimlane.

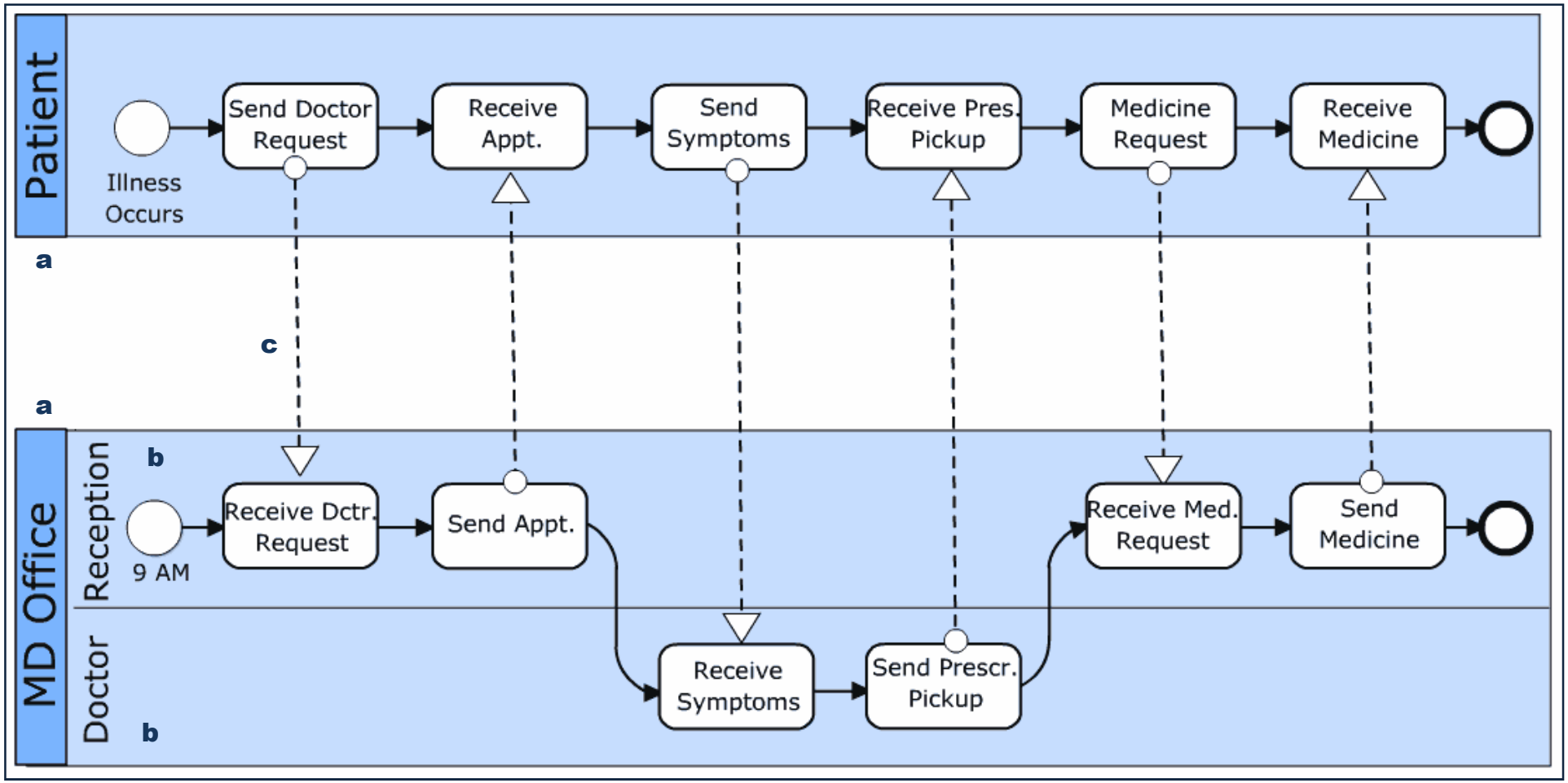


- In BPMN i processi privati sono quelli interni a una specifica organizzazione, ossia contenuti in un pool, e che non interagiscono con pool esterni (*orchestrazione*). Una collaborazione è una interazione sincronizzata di più processi senza un controllo centrale (*coreografia*), attraverso processi *pubblici* e message flows[1] (rappresentati con una freccia bianca e tratteggiata) tra pool (a).



- Sono **proibiti** flussi di controllo tra pool (b) flussi di messaggi nello stesso pool (c)

- Nell'esempio di seguito si ha una coreografia tra medico e paziente. Abbiamo due *pool*[a] (vasca) e due *lane*[b] (corsie), ciascuna rappresentante un sotto-responsabile o una sotto-unità.

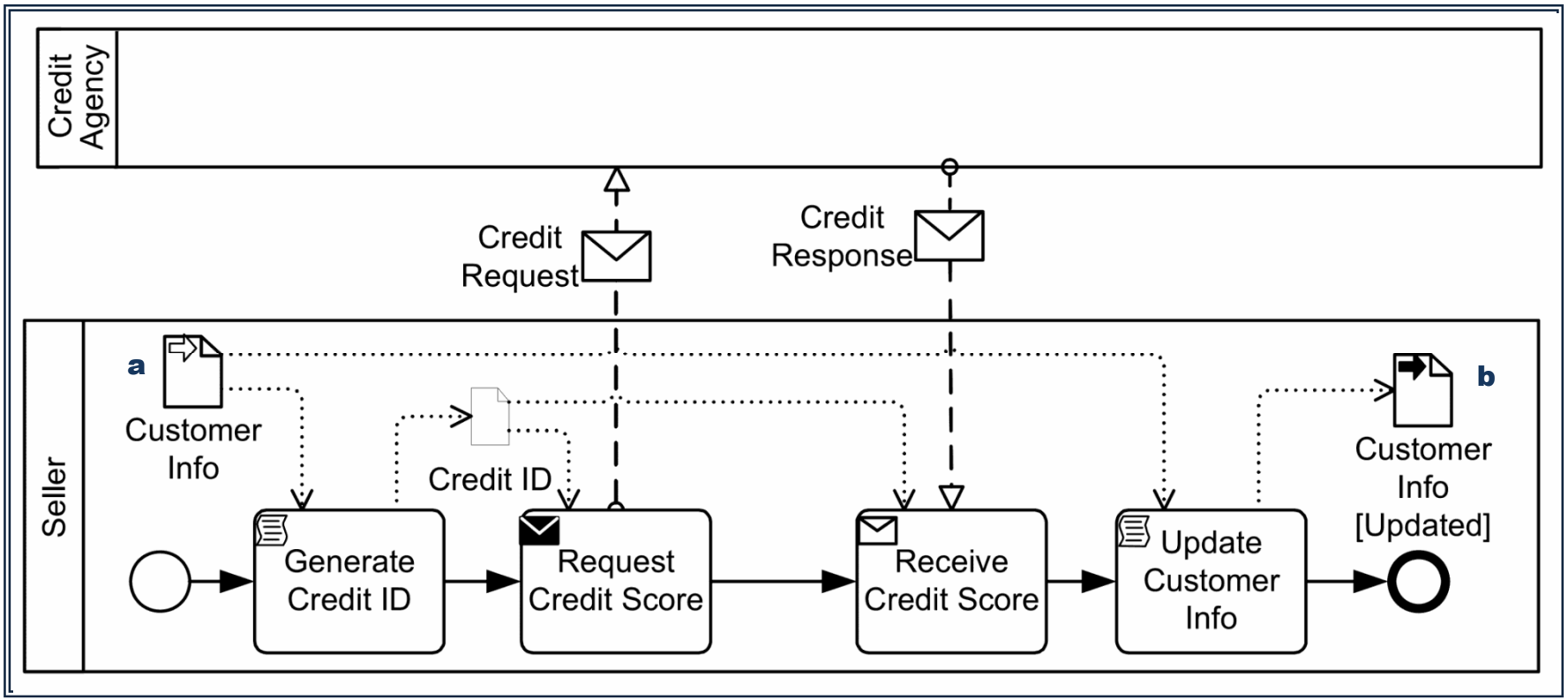


http://www.iet.unipi.it/m.cimino/ixl/res/mov11.swf

- Descrizione in linguaggio naturale:

```
 1 La Reception del centro medico apre alle 9.00
 2 Il Paziente avverte un malessere
 3 Il Paziente richiede un medico alla Reception
 4 La Reception riceve la richiesta di medico dal Paziente
 5 La Reception comunica un appuntamento al Paziente
 6 Il Paziente riceve l'appuntamento dalla Reception
 7 Il Paziente comunica i sintomi al Dottore
 8 Il Dottore riceve i sintomi dal Paziente
 9 Il Dottore prescrive la ricetta al Paziente
10 Il Paziente riceve la prescrizione dal Dottore
11 Il Paziente richiede le medicine alla Reception
12 La Reception riceve la richiesta di medicine dal Paziente
13 La reception fornisce le medicine al Paziente e termina.
14 Il Paziente riceve le medicine dalla Reception e termina.
```

- Il BPMN permette la modellazione complessiva del flusso di dati, attraverso i *data objects*, *messages* e *data store*.
- Una freccia con tratto punteggiato indica se il flusso dati è di ingresso o uscita e a/da quale attività. Data input[a] e data output[b] sono indicati anche da una freccia vuota e piena, rispettivamente.
- Esercizio: gestione delle richieste di credito da parte di un rivenditore (es. di automobili, elettrodomestici, ecc.).



- Descrizione in linguaggio naturale

1. Inizio
2. Il Commerciante genera un Id di credito tramite info sul Cliente
3. Il Commerciante richiede il computo del credito all'Agenzia
4. L'Agenzia riceve la richiesta di credito
5. L'Agenzia invia la risposta con il computo del credito
6. Il Commerciante riceve il computo del credito
7. Il Commerciante aggiorna le info sul Cliente, emettendo una scheda
8. Fine.

- Per essere completa, la descrizione dovrebbe includere anche altri aspetti (es. menzionare la tipologia di task)

- Tipi di task e icone in BPMN:
  - ✓ un *service task* è una funzione automatizzata (es. “calcola data di imbarco”);
  - ✓ un *receive/send task* riceve/invia un messaggio (es. “ricevi una richiesta di preventivo”, “invia una offerta”);
  - ✓ uno *user task* attende un input da un utente attraverso una interfaccia (es. “manda l’ordine d’acquisto”;
  - ✓ in un *business rule task* sono applicate alcune regole di business (es. su sistema avviato dal motore di processi) per produrre risultati (es. “determina lo sconto”);
  - ✓ uno *script task* contiene istruzioni direttamente eseguite dal motore di processo (es. “elenca le istanze di processo”);
  - ✓ un *manual task* viene svolto senza supporto di tecnologie dell’informazione (es. “confeziona i beni”);
  - ✓ in un *abstract task* non viene definito un tipo (es. “crea l’ordine”);