

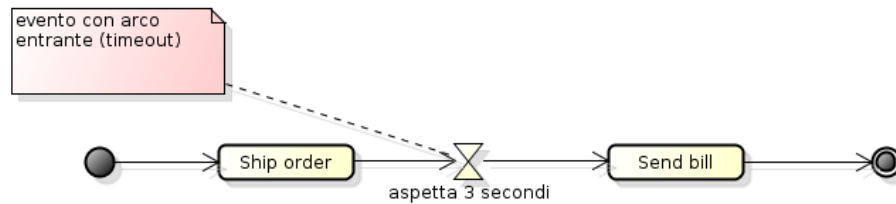
# **Lezione #13**

Lunedì, 11 Novembre 2013

*Christian Cardin, 09:30-11:15*

**Luca De Franceschi**

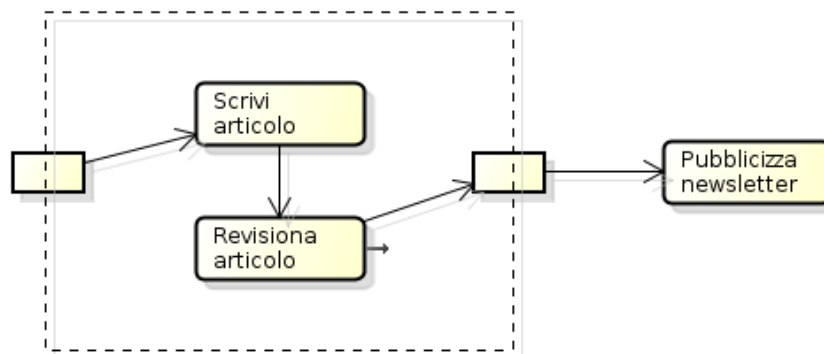
UML mette a disposizione gli **eventi esterni**, l'invio e la ricezione di messaggi a un altro diagramma di attività (quindi verso l'esterno). Quando voglio ricevere un messaggio dall'utente l'elaborazione viene bloccata in attesa della ricezione. Queste due primitive individuano il concetto di **sincronizzazione**. La *clessidra* è un **timer**, verifica il passaggio del tempo.



Gli archi che connettono due azioni servono per generare un flusso da un'azione a un'altra. E' possibile spezzare un arco in due con una notazione sotto forma di etichetta. Possiamo anche volendo passare un oggetto da un'azione a un'altra.



Le **regioni di esposizione** sono utili quando bisogna fare delle azioni su delle collezioni. Ogni elemento della lista è un *token*, un solo token in uscita dalla regione.

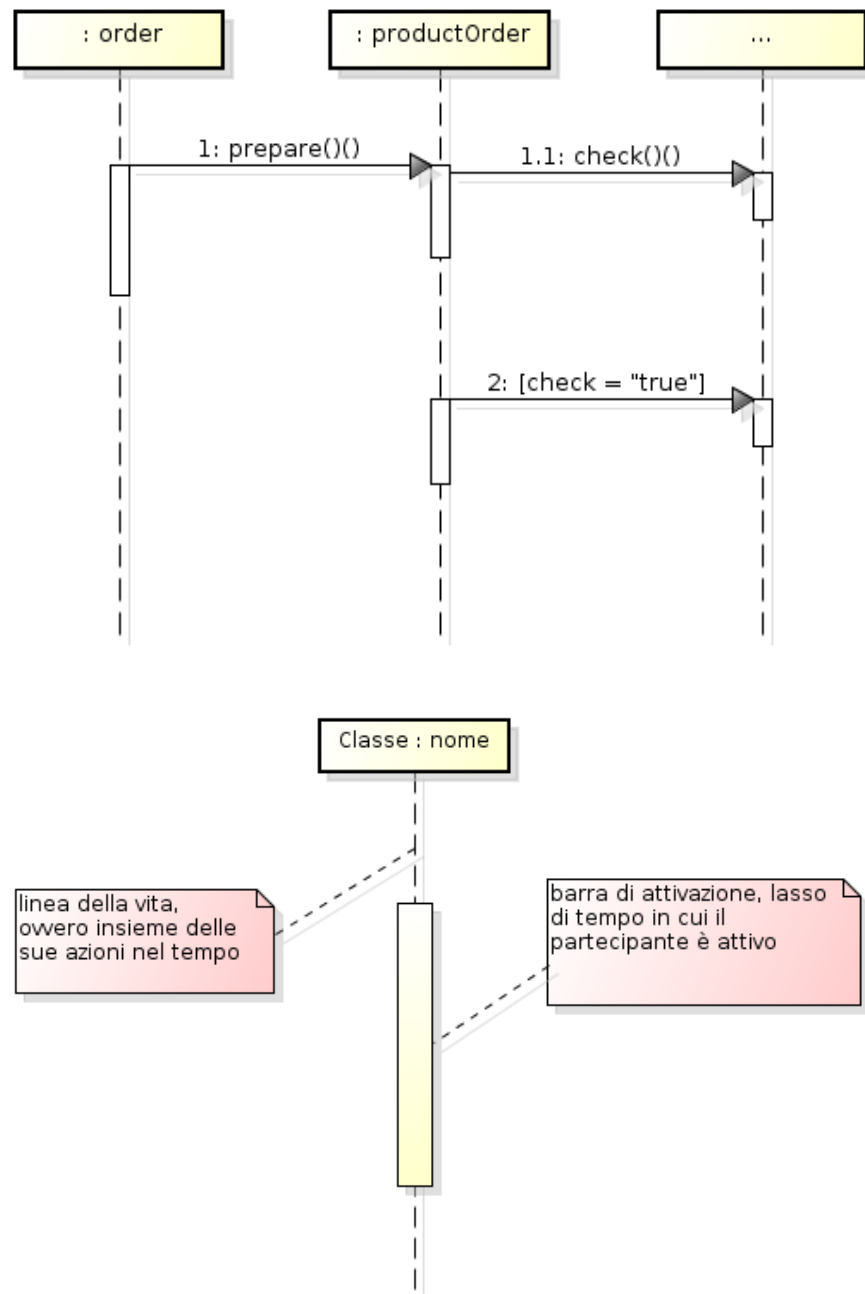


Non tutti i flussi possono arrivare alla fine, in tal caso può esserci un **nodo di terminazione** che fa morire un token. I diagrammi di attività sono i diagrammi più utili di tutti. Da utilizzare:

- Espressione di flussi paralleli;
- Per descrivere casi d'uso o requisiti direttamente dal capitolato tecnico.

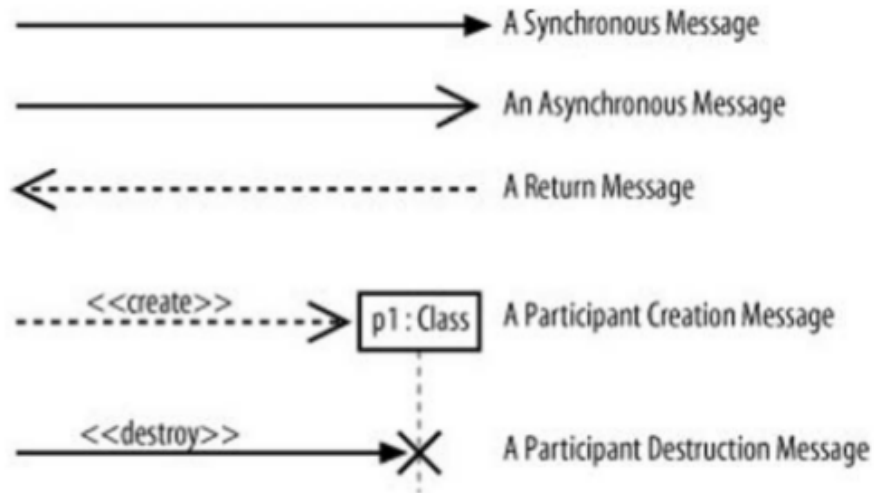
## Diagrammi di sequenza

Capire come due attori interagiscono tra di loro, interazione dinamica nel tempo di due parti. Si usano nella parte di definizione architetturale. Permette a chi dovrà andare a programmare di capire l'architettura e come far collaborare più gruppi di oggetti.

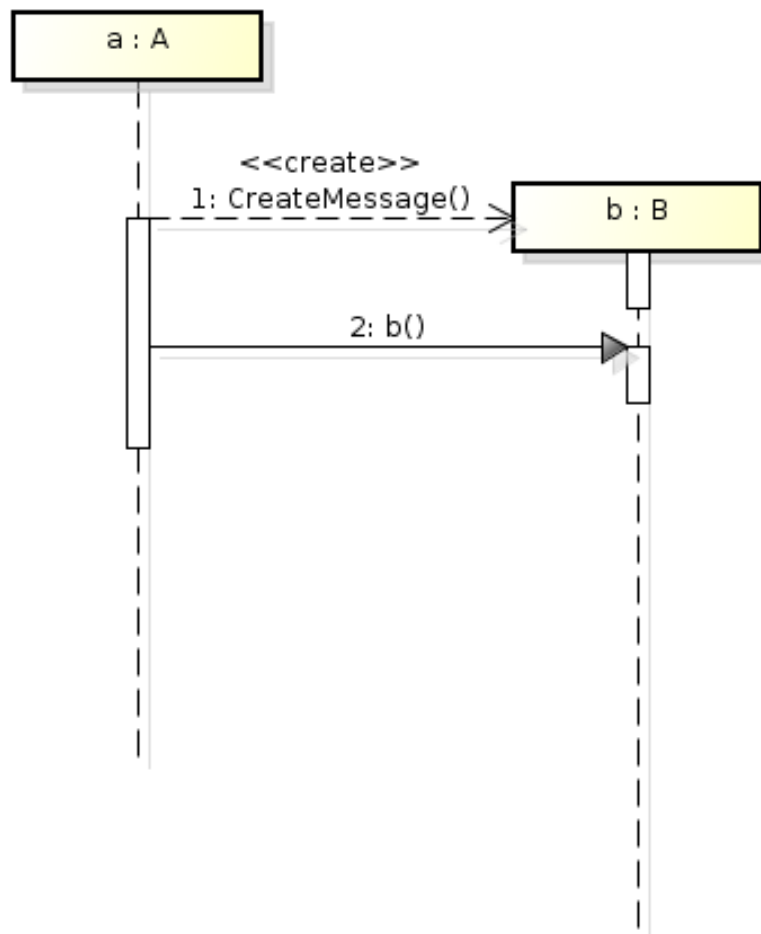


E' interessante vedere come avvengono le interazioni tra più partecipanti. Queste avvengono tramite **messaggi**. L'inizio di un messaggio si rappresenta con una freccia continua da un partecipante all'altro. E' possibile avere anche un messaggio che arriva dall'**esterno** del partecipante. L'evento esterno attiva il dia-

gramma di sequenza e lo fa partire. Il ritorno di un messaggio è rappresentato da una freccia tratteggiata. Esistono due possibili messaggi: **sincroni** e **asincroni**.

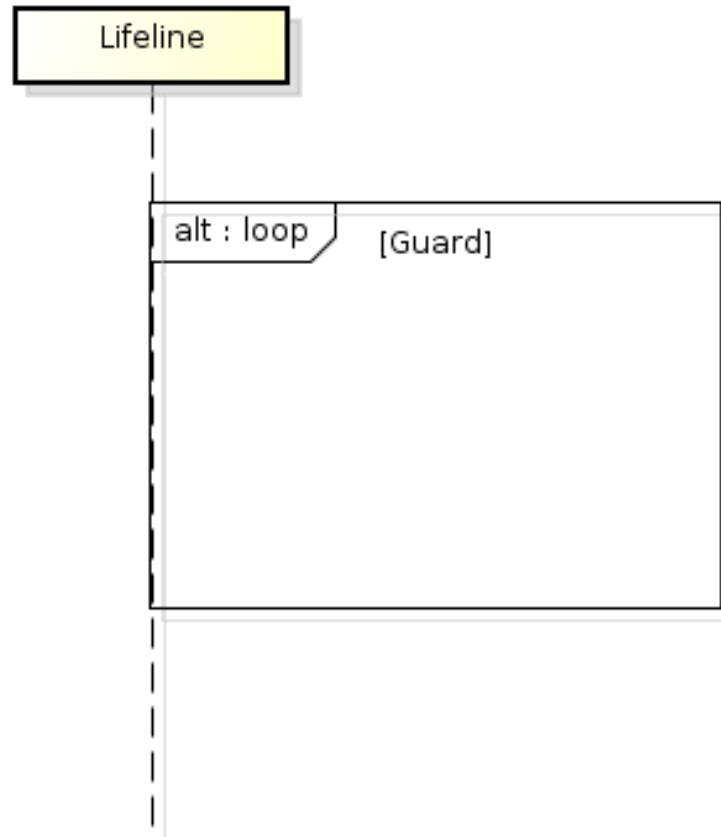


Un partecipante può creare un altro partecipante (normalmente tramite la *new*), per questo si usa la **create**:



Di contro un oggetto può richiedere la distruzione di un altro oggetto. In questo caso si invia un altro messaggio e si mette una X sulla linea della vita del partecipante. Un oggetto può anche suicidarsi, ovvero autodistruggersi.

In UML2 hanno introdotto le primitive per rappresentare i **cicli** e le **condizioni** tramite i *frame* di interruzione, ciascuno con un'etichetta che rappresenta qualcosa.



Ci sono vari **operatori** (etichette):

- **ALT;**
- **OPT;**
- **PAR;**
- **LOOP;**
- **REGION;**
- **NEG;**
- **REF;**
- **SD.**

Se ho bisogno di modellare la collaborazione uso i diagrammi di sequenza, se devo modellare algoritmi uso i diagrammi di attività.

Controllo **centralizzato** e **distribuito**. Bisogna cercare di delegare il più possibile agli oggetti le operazioni su se stessi (*delegation pattern*). Nel contratto distribuito non abbiamo più un registro che *schedula* tutte le attività, ma tutto parte da un singolo oggetto che invoca metodi su altri oggetti, manda e riceve messaggi.

**Esercizio:**

