# Sviluppo di sistemi su vasta scala con RUP (Rational Unified Process)

Maria Ericsson

White paper del software Rational

TP 156

Rational®
the software development company

# Indice

## Cronologia

Questo documento è tratto da "Systems of Interconnected Systems", pubblicato in ROAD, maggio-giugno 1995, by Ivar Jacobson, Karin Palmkvist e Susanne Dyrhage [1]. Il documento prende spunto dai molti progetti di sviluppo di sistemi su vasta scala, e vuole essere in linea con la versione 5.1 di RUP (Rational Unified Process) [2] e con UML (Unified Modeling Language) [3].

## Sistemi di sistemi interconnessi

Quando si sviluppano sistemi su vasta scala la complessità aumenta in maniera considerevole. Non solo è necessaria la capacità di comprendere un insieme di artefatti più complessi, ma vengono introdotti eccessi a causa dell'esigenza di gestire un'insieme più vasto
di risorse. Questo saggio descrive un pattern strutturale utilizzato per
controllare l'eccesso di complessità aggiunta. Il pattern strutturale, tra l'altro descritto in [4], viene definito come sistema di sistemi interconnessi.

Tale costruzione è utile nella creazione di sistemi molto vasti o complessi, come quelli di comando e controllo o
soluzioni IT altamente integrate. Tali tipi di "super sistemi" vengono nella maggior parte dei casi divisi in diverse parti separate, ciascuna sviluppata
indipendentemente come sistema autonomo. Un super sistema è implementato da una serie di sistemi interconnessi comunicanti per soddisfare le esigenze del super sistema. Uno di questi sistemi rappresenta le capacità generali e viene definito sistema superordinato.Gli altri rappresentano un parte dell'insieme e possiamo definirli sistemi subordinati. Un sistema superordinato è nettamente diviso da quelli subordinati che lo implementano. La relazione tra i
diversi tipi di sistemi è chiara: dalla prospettiva dei sistemi superordinati, quelli subordinati sono sottosistemi,
vedere la figura 1.

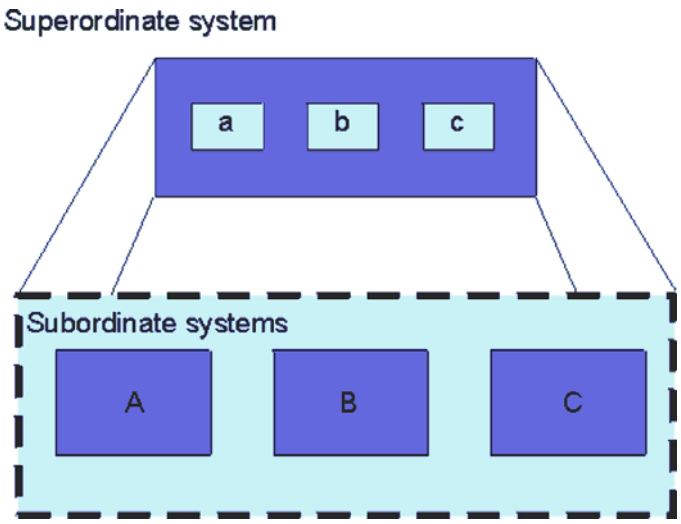


Figura 1. La specifica di un sistema superordinato è implementata da un sistema di sistemi interconnessi, nel quale i sistemi A, B e C sono implementazioni dei sottosistemi del sistema superordinato rispettivamente a, b e c.

Separare il sistema superordinato dai relativi sistemi subordinati comporta diversi vantaggi:

- I sistemi subordinati possono essere gestiti separatamente durante tutte le attività del ciclo di vita, compreso vendita e distribuzione.
- Utilizzare un sistema subordinato per implementare altri sistemi superordinati può diventare più facile collegandolo ad altri sistemi di sistemi interconnessi.

- Quando si inizia a costruire un sistema, non sempre si sa se si tratta di un sistema di sistemi interconnessi o meno. Si può cominciare con una vista di sistema "semplice" per poi determinare in seguito se sarà necessario applicare il pattern di sistemi di sistemi interconnessi o meno.
- Ciò consente di effettuare modifiche interne ai sistemi subordinati senza sviluppare una nuova versione del sistema superordinato. Nuove versioni saranno richieste solo in seguito a modifiche funzionali superiori.

Ogni sistema subordinato è associato ad un insieme di artefatti, con una chiara tracciabilità tra loro. Esiste anche una tracciabilità tra l'insieme di artefatti dei sistemi subordinati e i corrispondenti insieme di artefatti del sistema superordinato. Ciascun sistema subordinato può essere gestito come progetto di sviluppo separato con proprie fasi di ciclo di vita: Inizio, Elaborazione, Costruzione e Transizione.

Se il "super sistema" che si sta costruendo è molto ampio, un sistema subordinato può essere suddiviso ulteriormente ed essere così trattato come un sistema di sistemi interconnessi.

## Il ciclo di vita dello sviluppo del software

In RUP, il ciclo di vita dello sviluppo è presentato e discusso sotto due prospettive: gestione e sviluppo, vedere la figura 2.

Dalla prospettiva della gestione, si passa attraverso quattro fasi del ciclo di vita per sviluppare un sistema o una nuova generazione di un sistema.

Dalla prospettiva dello sviluppo, vengono sviluppate in modo iterativo versioni del sistema che sono sempre più complete. Le attività che si svolgono durante l'iterazione, sono state raggruppate in RUP in un insieme di flussi di lavoro di base, ciascuno dei quali si focalizza sulla descrizione di alcuni aspetti del sistema, dando luogo ad un modello del sistema o ad una serie di documenti.

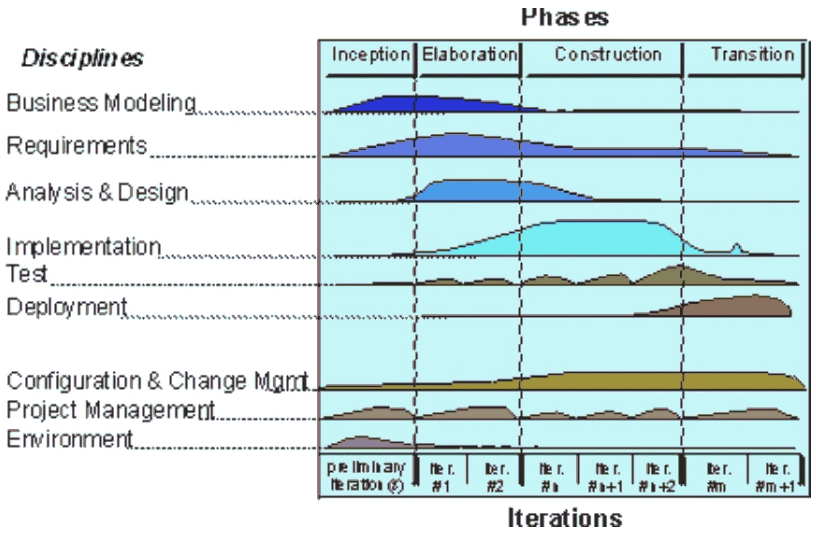


Figura 2. Il modello iterativo

Applicando la stessa regola ai sistema di sistemi interconnessi, il sistema superordinato e ciascuno dei relativi sistemi subordinati hanno cicli di vita propri e sono spesso trattati come progetti autonomi.

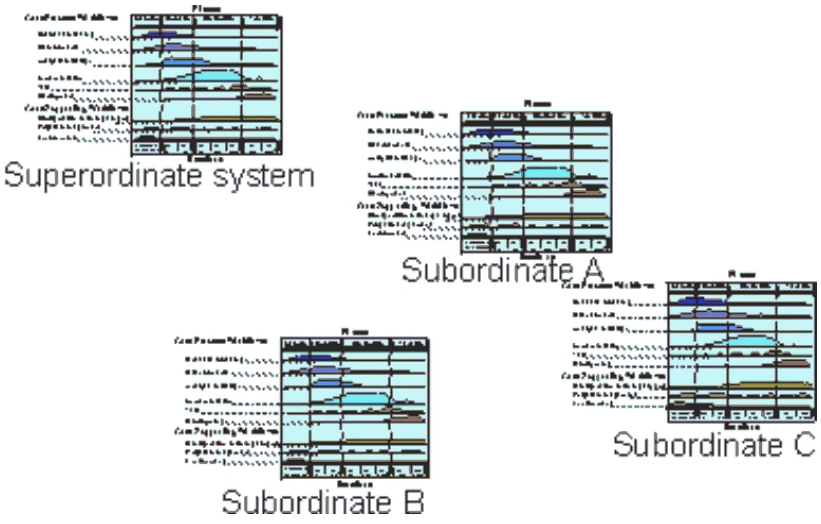


Figura 3. Ciascun sistema, superordinato e subordinato, ha cicli di vita propri.

I cicli di vita ovviamente hanno delle dipendenze, gestirle in modo corretto è una delle sfide nello sviluppo di un sistema di sistemi interconnessi. Le dipendenze sono dei seguenti tipi:

- Sono dipendenti nel tempo, i cicli di vita del sistema superordinato inizia prima. Quando il sistema superordinato ha passato almeno una iterazione e le interfacce dei sistemi subordinati sono relativamente stabili, i cicli di vita dei sistemi subordinati possono partire. Infatti, si potrebbe ignorare di quali sistemi subordinati si tratti addirittura finché non si sia passata almeno un'iterazione nel sistema superordinato.

- Il ciclo di vita del sistema superordinato può essere gestito una volta che le interfacce di sistemi subordinati sono stabili. Ciò significa che non è stato fatto nessun sviluppo attivo, solo se sorgono problematiche che richiedono modifiche alle interfacce dei sistemi subordinati.

- Le interfacce dei sistemi subordinati sono gestite da coloro che sviluppano il sistema superordinato. Per ulteriori informazioni sulle interfacce, vedere [3] e [5].

- Le classi che implementano le interfacce dei sistemi subordinati sono gestite da coloro che sviluppano i sistemi subordinati.

## Workflow e artefatti dello sviluppo di sistema

Una supposizione naturale è che il sistema superordinato, così come i sistemi subordinati, possono essere sviluppati con lo stesso insieme di artefatti e attraverso lo stesso flusso di lavoro dei sistemi normali e non composti. Prima di procedere nel dimostrare come ciò può essere eseguito, devono essere introdotti gli artefatti e i workflow. Nel Rational Unified Process, vengono introdotti cinque flussi di lavoro del processo, vedere figura 4, che sono:

- Progettazione del business, lo scopo è di stabilire l'organizzazione con cui il sistema verrà utilizzato, per comprendere meglio le esigenze e i problemi da risolvere nel sistema. Il risultato è un modello di caso d'uso e di oggetto di business. Tale flusso di lavoro può essere considerato facoltativo. Se l'organizzazione con la quale il sistema deve essere eseguito è molto semplice, potrebbe non aggiungere valore.

- Requisiti, con l'obiettivo di catturare e valutare i requisiti, concentrandosi sull'utilizzabilità. Questo dà luogo a un modello Caso d'uso, in cui attori che rappresentano unità esterne comunicano con il sistema, e casi d'uso che rappresentano sequenze di transazione, producendo risultati misurabili del valore degli attori.

 Analisi & progettazione, con lo scopo di analizzare l'ambiente di implementazione previsto e 'effetto che avrà sulla costruzione del sistema. Il risultato è un modello di oggetto (il modello di progettazione), che comprende realizzazioni del caso d'uso che mostrano il modo in cui comunicano gli oggetti per eseguire il flusso del caso d'uso. Dovrebbero essere comprese le definizioni di interfaccia per classi e sottosistemi, specificandone le responsabilità in termini di operazioni realizzate. Questo modello di oggetto è adatto anche all'ambiente di implementazione in termini di linguaggio di implementazione, distribuzione ecc. Può essere a volte utile considerare i risultati dell'analisi come un modello separato, definito in seguito modello di analisi.

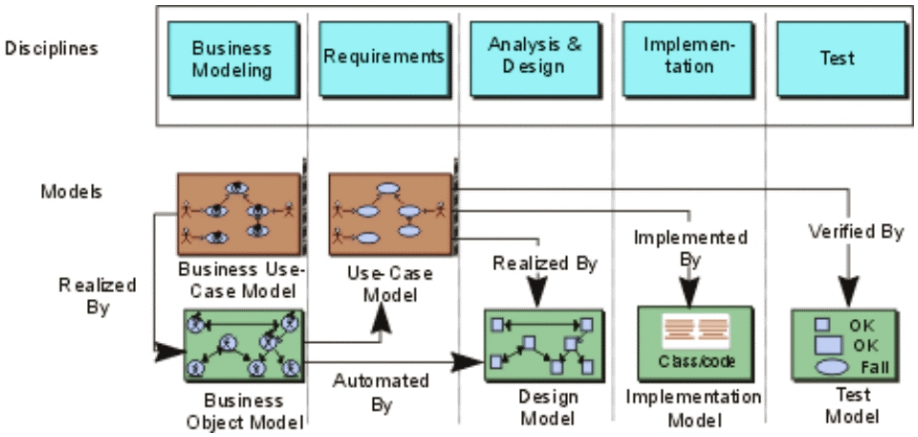


Figura 4. Ogni flusso di lavoro del processo di base è associato a particolari insiemi di modelli.

 Implementazione, con lo scopo di implementare il sistema in ambiente di implementazione prescritto. Il risultato è un codice di origine eseguibile e alcuni file.

 Test, con l'obiettivo di assicurare che il sistema è quello previsto e che non vi siano errori nell'implementazione. Ciò dà luogo a un sistema certificato pronto per la consegna.

## Sviluppo di un sistema di sistemi interconnessi

È necessario stabilire il modo in cui le responsabilità del sistema possono essere distribuite su diversi sistemi, ciascuno dei quali attento ad un sottoinsieme ben definito di tali responsabilità. Questo significa che l'obiettivo principale è definire le interfacce tra tali sistemi subordinati. Una volta terminato, il resto del lavoro può essere fatto separatamente per ogni sistema subordinato seguendo il principio della divisione e acquisizione. Questo è quanto c'è da fare per il sistema come insieme, a parte verificarlo una volta terminata l'implementazione.

### Criteri di scomposizione

Come decidere dunque se scomporre o no il sistema in un sistema di sistemi interconnessi? Bisogna considerare un paio di caratteristiche:

 Per un sistema di misura e complessità considerevole, è possibile dividere il problema in parti più piccole più semplici da comprendere.

 Si sta lavorando su sistemi fisicamente separati? In questo caso si tratta spesso di sistemi precedenti o di un'architettura precedente.

 La scomposizione favorisce una definizione di interfacce naturali e ristrette tra parti del sistema.

 Si può decidere di implementare una parte del sistema utilizzando alcuni prodotti COTS (Commercial-Off-The-Shelf) superiori. La scomposizione aiuterà a chiarire in che modo si vuole utilizzare tali prodotti.

- La partizione consente di ottenere una buona parte dell'organizzazione di sviluppo distribuita e di dividere equamente il lavoro tra team geograficamente dispersi.

I rischi da tenere in conto sono:

- Eccessivo utilizzo della scomposizione può nascondere il problema generale per tutti i dettagli.
- Utilizzando sistemi o team fisicamente separati si corre rischio di eliminare ogni forma di riutilizzo e ottenere con un sistema rigido.

### Organizzazione

Per ridurre i precedenti rischi, è importante avere un gruppo di persone con il compito di supervisionare l'intero sforzo di sviluppo.

Questo gruppo è in genere definito team di architettura e si concentrerà sulle seguenti questioni:

- Che ci sia un'architettura generale definita e che sia seguita nei sistemi subordinati.
- Che vi sia la giusta attenzione sul riutilizzo e sulla condivisione di esperienze tra i sistemi subordinati.
- Che vi sia una comprensione chiara di quali artefatti produrre e quali relazioni esistono tra quelli dei sistemi subordinati e superordinati.
- Che sia stata definita un'efficace strategia di gestione delle modifiche e che sia seguita da tutti i team.

Il team di architettura, può, anche se non sempre, gestire lo sviluppo del sistema superordinato. Per una discussione più approfondita sull'organizzazione, vedere [6].

### Il ciclo di vita del sistema superordinato

Per prima cosa, si può facoltativamente scegliere di fare la progettazione del business per comprendere meglio il contesto del sistema. Questo aggiunge valore se:

- gli sviluppatori devono comprendere meglio l'organizzazione,
- questa è eterogenea nel modo di fare e la terminologia e i processi devono essere allineati
- lo sforzo di progettazione di software è fatto in congiunzione con un sforzo di riprogettazione del business.

Vedere anche [6].

Tale sforzo ha come conseguenza un modello Caso d'uso e Oggetto di business. In alternativa si può scegliere di fare una progettazione del business limitata, solo guardando ai concetti chiave nel campo di dominio e documentandoli in un modello di oggetto di business. Quest'attività viene spesso definita modellazione di dominio.

Avendo disponibili tutti i modelli di business necessari, si deve iniziare a scegliere i requisiti sull'intero sistema.
La modellazione di requisiti per un sistema di sistemi interconnessi è necessaria quanto per altri sistemi. Un modello Caso d'uso è un modo molto naturale per esprimere i risultati, vedere [7]. Il modo più diretto per verificare questi modelli Caso d'uso è di presupporre che catturi completamente i requisisti comportamentali del sistema. Raramente questo può essere il caso. Poiché è necessario implementare il sistema con altri sistemi, il sistema generale è probabilmente abbastanza complesso, non è quindi una buona idea provare ad essere esaurienti a questo livello. Un modello Caso d'uso superordinato da un quadro completo e semplificato dei requisiti funzionali del sistema. Non è necessario essere troppo dettagliati a questo livello, in quanto la modellazione dettagliata sarà eseguita in ciascun sistema subordinato di implementazione. È anche vero che molti requisiti non sono necessariamente visibili in un caso d'uso superordinato che intersecano vari sottosistemi. Tali requisiti si possono definire "locali" ad un sottosistema.

Lo scopo di analisi & progettazione è di raggiungere un'architettura robusta del sistema, ovviamente di vitale importanza per un sistema di sistemi interconnessi. Gli sviluppatori del sistema superordinato devono ottenere una struttura robusta dei sistemi subordinati, mentre non devono avere a che fare con le strutture interne. Si modellerà dunque una divisione del sistema in parti più piccole utilizzando sottosistemi. Per ottenere la giusta serie di sottosistemi e per farsi una prima idea di come distribuire le responsabilità del sistema superordinato su questi sottosistemi, si svilupperà un modello Analisi. Le classi di analisi dovrebbero rappresentare i ruoli svolti dalle cose nel sistema quando vengono eseguiti casi d'uso ad alto livello. Il modello Analisi dunque da un quadro semplificato della struttura completa dell'oggetto, in analogia con il modello Caso d'uso ad alto livello.

Le classi correlate funzionalmente vengono raggruppate in sottosistemi. È possibile quindi ottenere una struttura di sottosistemi ideale, nel senso che è basata solo su criteri funzionali, ad esempio non abbiamo preso in considerazione la distribuzione dei requisiti. Altro fattore altamente influente è l'esistenza di sistemi precedenti, che potrebbero adempiere a qualcuna delle responsabilità definite nel modello di analisi. L'esistenza di tali sistemi può causare una ripartizione delle responsabilità così da ottenere la massima riutilizzabilità delle capacità esistenti.
Il risultato della progettazione potrebbe essere una struttura di sottosistema molto differente da quella definito sulla base di criteri funzionali durante l'analisi. Si ottiene così una struttura di sottosistemi di progettazione, che saranno implementati da un sistema subordinato, vedere la figura 5. Per essere in grado di continuare il lavoro d i sviluppo per ciascun sistema separatamente, le interfacce vengono definite per ogni sottosistema. Infatti, la definizione di interfacce è l'attività più importante eseguita a livello superordinato, in quanto queste forniscono regole per lo sviluppo dei sistemi subordinati. Le classi di progettazione non sono definite, la sola cosa da fare è definire le interfacce dei sottosistemi di progettazione.

Non viene attuata nessuna implementazione come parte del ciclo di vita del sistema superordinato, piuttosto che esplorare particolari aspetti tecnici del sistema.

Il workflow finale è il test, che in questo caso significa test di integrazione quando vengono assemblati i diversi sistemisubordinati e anche per verificare che ogni caso d'uso superordinato è eseguito secondo le specifiche dai sistemi interconnessi nella cooperazione.

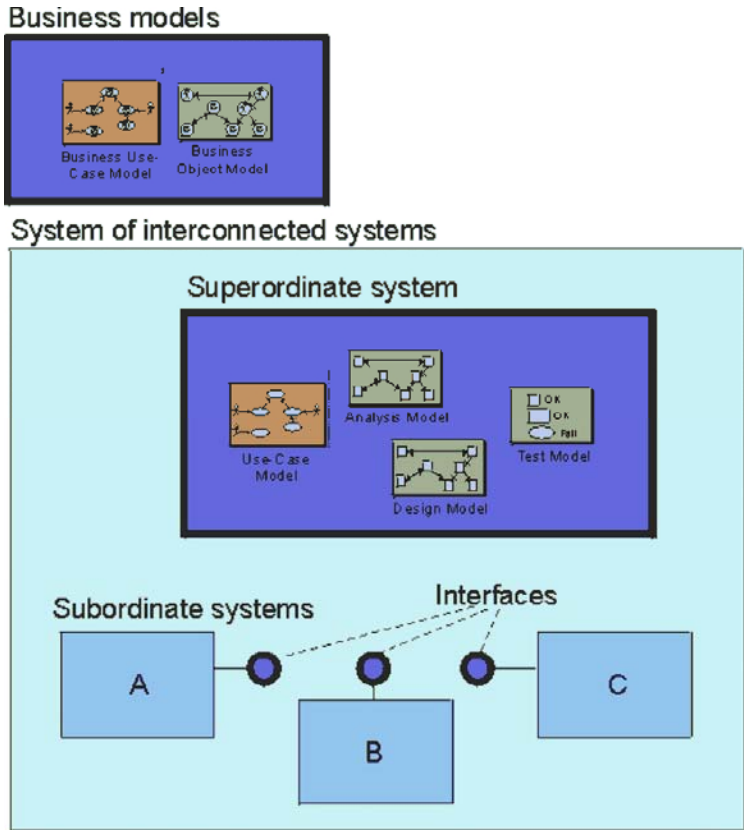


Figura 5. Il sistema superordinato è descritto da una serie di modelli, mentre i sottosistemi sono definiti nel modello di progettazione ad alto livello saranno implementati dai sistemi subordinati. Le interfacce ai sistemi subordinati sono gestiti dal sistema superordinato.

Per mostrare come si lavorerebbe con il sistema superordinato, sono riportati vari esempi di piani di iterazione, uno per un'iterazione nella fase Inizio del ciclo di vita del sistema superordinato, uno per un'iterazione nella fase Elaborazione. Sono utilizzati diagrammi dell'attività per descrivere i piani d'iterazione. Gli stati d'azione in questi diagrammi corrispondono ai dettagli del workflow come definito nel Rational Unified Process.

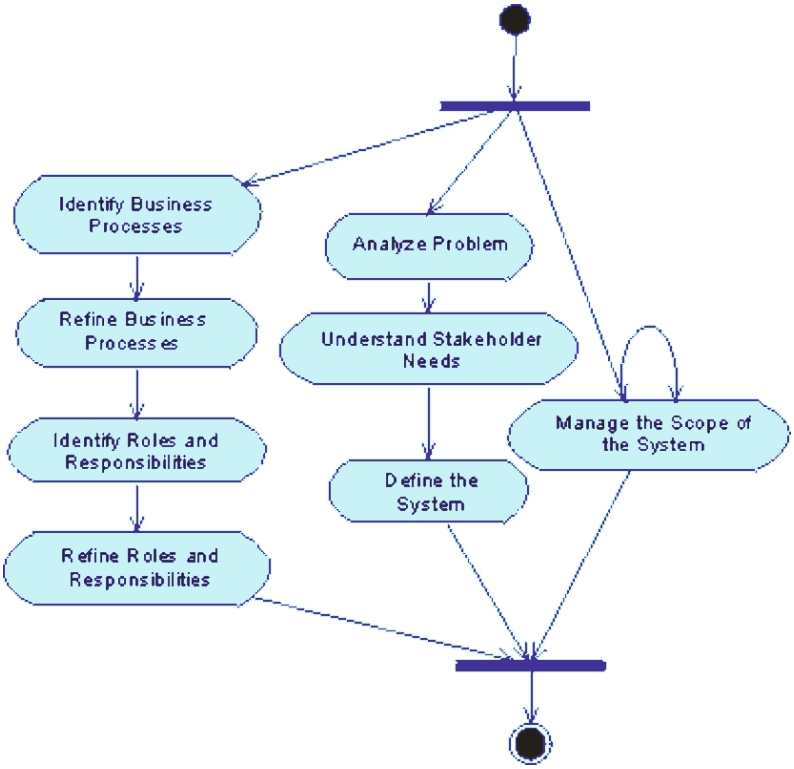


Figura 6. Un diagramma dell'attività che descrive un esempio di un piano di iterazione iniziale relativo al sistema superordinato.

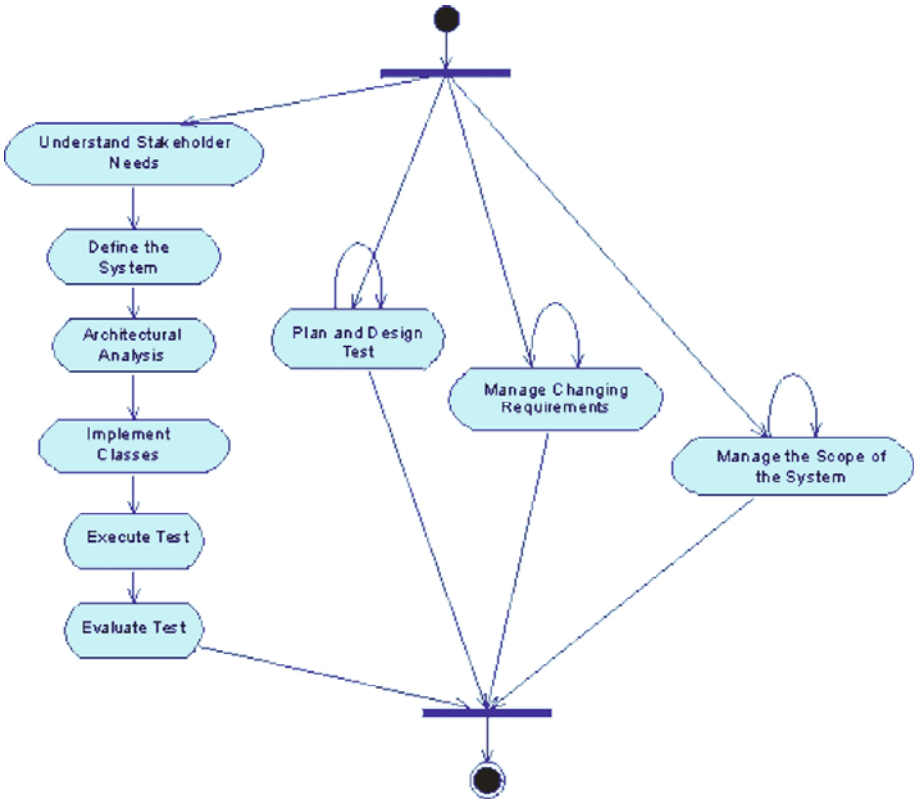


Figura 7. Un diagramma dell'attività che descrive un esempio di un piano di iterazione di elaborazione relativo al sistema superordinato. È presente lo stato d'azione "implementazione di classi" perché si potrebbero fare delle implementazioni limitate di prototipi per esplorare aspetti tecnici del sistema.

## Il ciclo di vita del sistema subordinato

Ogni sistema subordinato è sviluppato nel modo comune, come una scatola nera che consideri altri sistemi con cui comunica come attori. Si eseguirà il solito insieme di attività e si svilupperà lo stesso gruppo di modello, come scritto in precedenza, per ciascun sistema. Se i modelli a livello superordinato sono definiti nei dettagli, si ottiene una completa ricorsività tra i modelli a livelli differenti, ma, come detto prima, raramente questo può essere il caso.

Per il sistema subordinato, si eseguirà il flusso di lavoro dei requisiti. Le interfacce e i casi d'uso del sistema superordinato saranno l'input primario per comprendere i legami del sistema subordinato e chi sono i relativi attori.

Nell'attuare analisi & progettazione per il sistema subordinato, le interfacce definite nel sistema superordinato saranno le "condizioni necessarie", insieme ai casi d'uso d'alto livello.

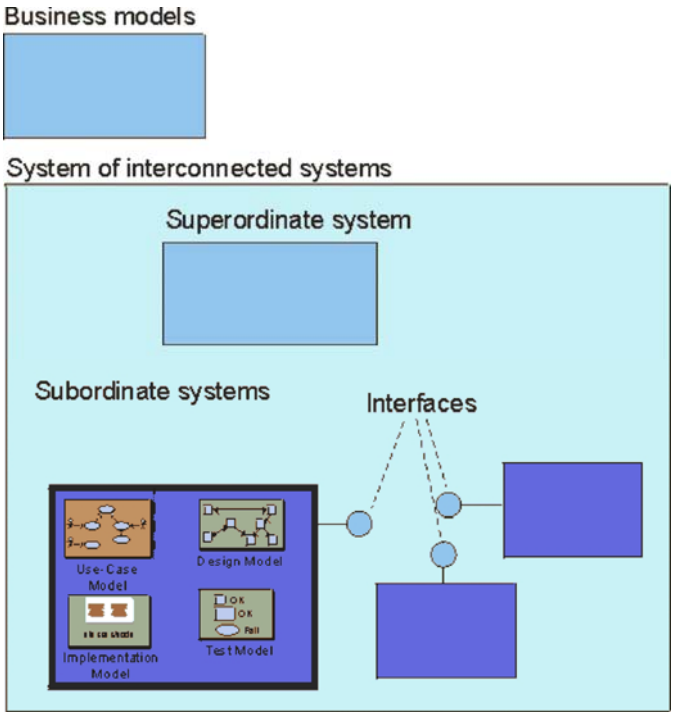


Figura 8. I sistemi subordinati sono descritti dal proprio insieme di modelli.

Per mostrare come si lavorerebbe con il sistema subordinato, sono riportati due esempi di piano d'iterazione del ciclo di vita .

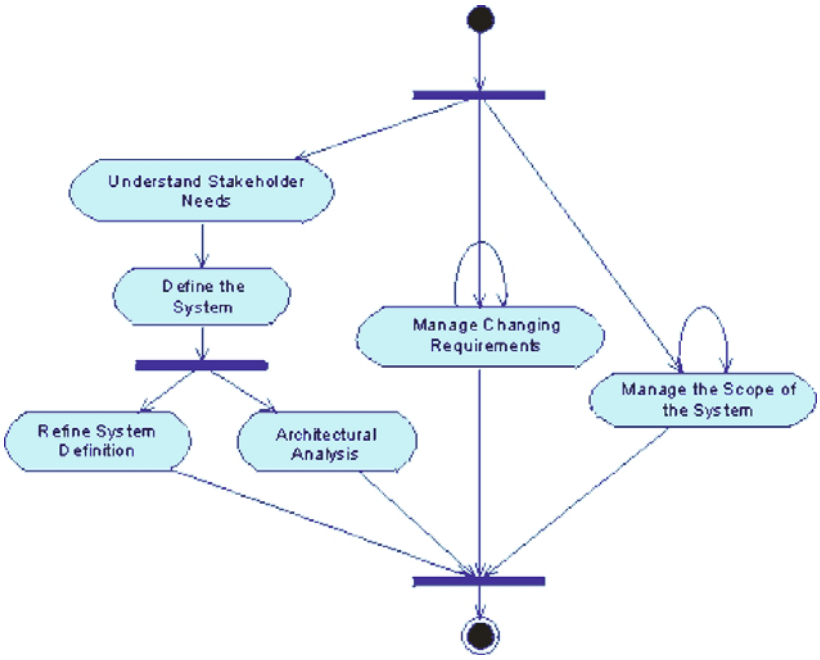


Figura 9. Un esempio del piano di iterazione iniziale relativo al sistema subordinato. Si tratta di un'iterazione incompleta, in quanto non viene prodotto nessun elemento eseguibile.

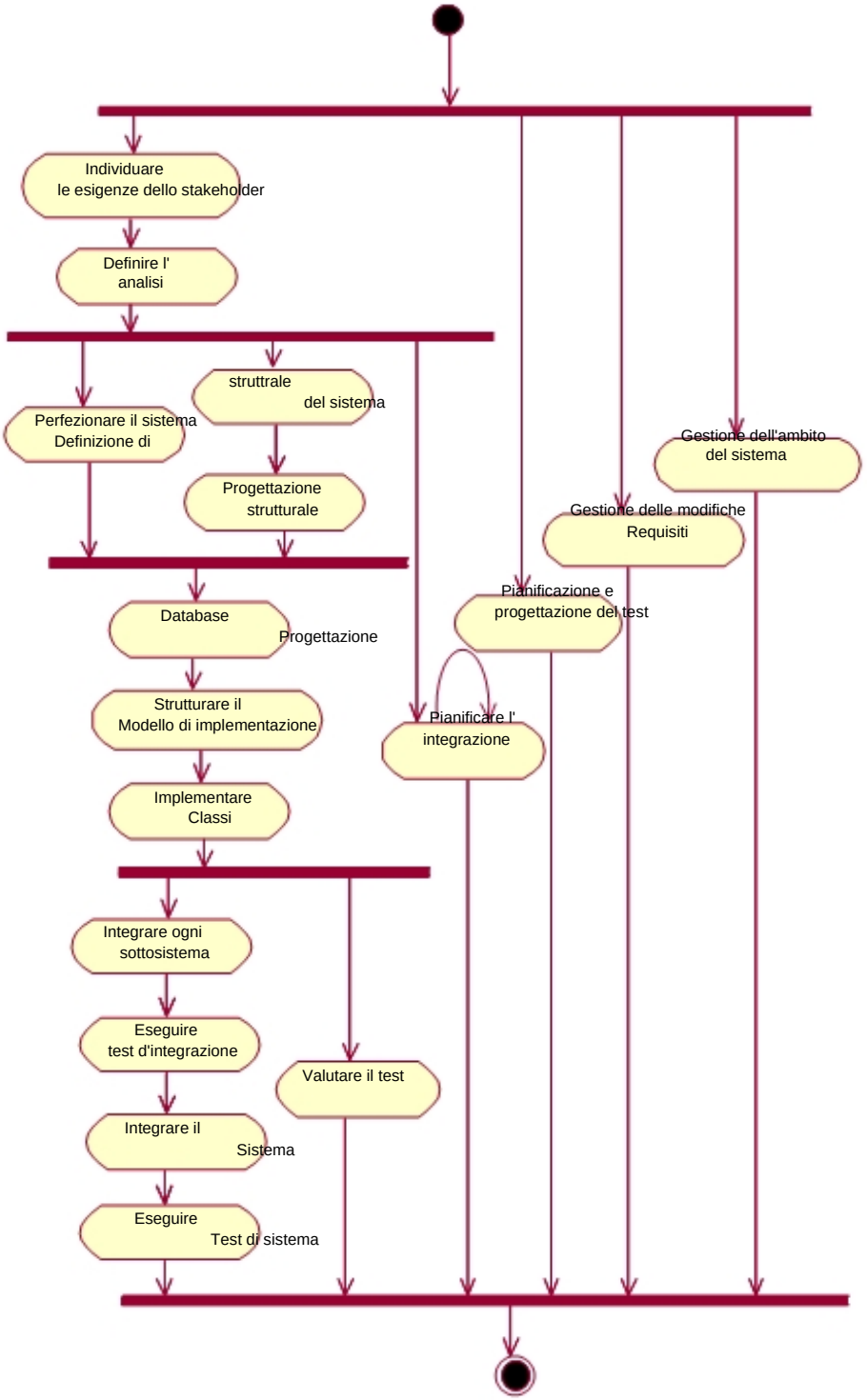


Figura 10. Un esempio di piano di iterazione di elaborazione per il sistema subordinato. L'attenzione nell'elaborazione è concentrata sul completamento delle definizione del sistema perfezionato e dell'architettura.

## Casi d'uso nei sistemi di sistemi interconnessi

Si dovrebbe costruire un modello Caso d'uso per ognuno dei sistemi, sia superordinati che subordinati, nei sistemi di sistemi interconnessi. Questi sono dipendenti nel modo seguente (vedere anche la figura 11):

- Un caso d'uso ad alto livello nel sistema superordinato viene diviso (non sempre ma spesso) in sottosistemi. Ciascun 'pezzo' diventa un caso d'uso nel modello per il sistema subordinato, vedere la figura 11.
- Dalla prospettiva di un sistema subordinato, gli altri sistemi subordinati sono attori in proprio modello di caso d'uso, vedere la figura 12.

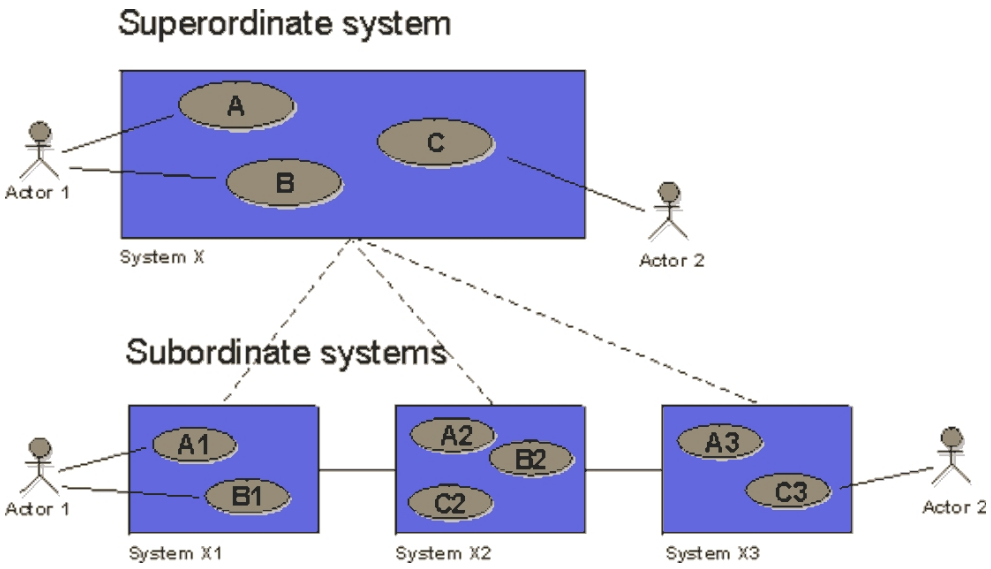


Figura 11. La relazione tra caso d'uso ad alto livello nel sistema superordinato e casi d'uso dettagliati nei sistemi subordinati.

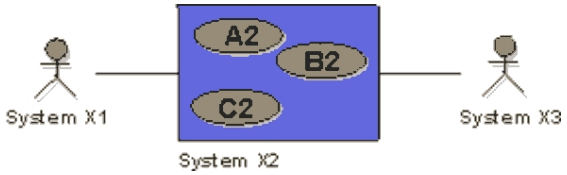


Figura 12. Nel modello di caso d'uso del sistema subordinato X2, gli altri sistemi subordinati X1 e X3 vengono visti come attori.

Vi sono alcune considerazioni speciali per i casi d'uso che descrivono il sistema superordinato. Poiché in un certo senso verranno ridescritti tutti i requisiti per ogni sistema subordinato, non vi è motivo di approfondire l'argomento troppo in questi casi d'uso. In un caso normale, è sufficiente in genere annotare la struttura passo per passo al flusso di eventi nei casi d'uso ad alto livello e non scriverla dettagliatamente in un testo narrativo.

In questo modello di caso d'uso non si dovrebbero utilizzare le relazioni tra i casi d'uso (generalizzazione, estensione, inclusione). In generale, non aggiunge valore per le seguenti ragioni:

- Non saranno descritti i casi d'uso ad alto livello in dettagli, dunque non è necessario preoccuparsi del testo che appare in vari posti.

 Ad ogni modo saranno strutturate le informazioni nel momento in cui si "dividono" casi d'uso ad alto livello in sistemi subordinati.

Unire questo a altri meccanismi di strutturazione può creare confusione.

Esiste una importante eccezione, cioè il caso in cui si mira a individuare componenti riutilizzabili nel sistema di sistemi interconnessi. Strutturare il modello di caso d'uso superordinato per individuare casi d'uso generici è un modo efficace per trovare componenti riutilizzabili. Vedere [6] per ulteriori dettagli su quest'argomento.

## Modelli di progettazione nei sistemi di sistemi interconnessi

Ogni sistema nel proprio sistema di sistemi interconnessi, sia superordinati che subordinati, dovrebbe avere un proprio modello di progettazione. Questi ultimi sono collegati nel modo seguente:

 Sottosistemi nel modello di progettazione per il sistema superordinato definisce i limiti dei sistemi subordinati.

 Le operazioni definite sui sottosistemi del sistema superordinato costituiscono un input per definire le interfacce ai sistemi subordinati.

Il modello di progettazione del sistema superordinato è descritto in modo meno dettagliato dei modelli di progettazione subordinate. Si produce quanto segue:

 Sottosistemi, descritti brevemente.

 Realizzazioni del caso d'uso, in termini di quanto collaborano i sottosistemi. Il modo migliore di documentare queste realizzazioni di casi d'uso ad alto livello è disegnare diagrammi di sequenza. È producendo tali diagrammi che sarà possibile definire la "divisione" di casi d'uso ad alto livello in sistemi subordinati, vedere la figura 13.

 Operazioni dei sottosistemi.

 Definizioni di interfaccia per i sottosistemi.

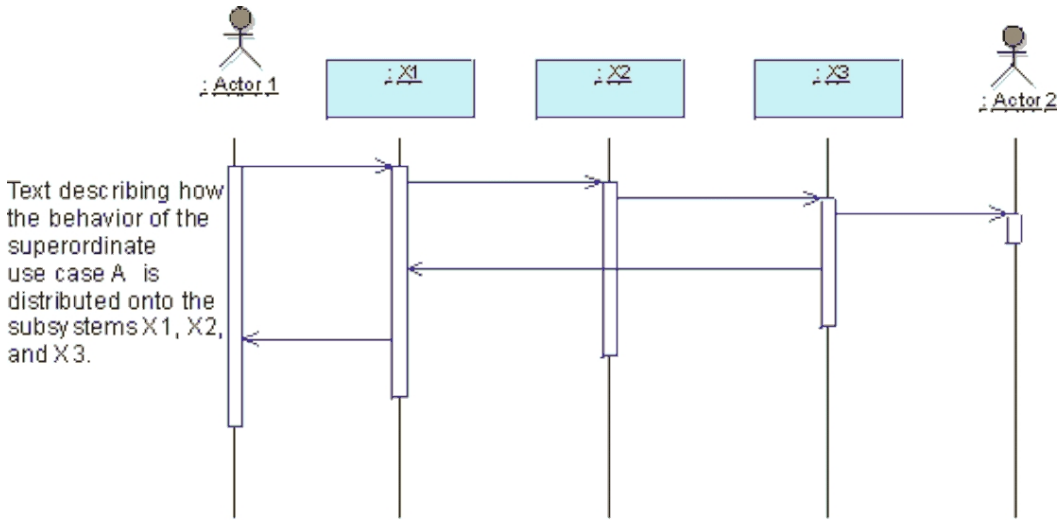


Figura 13. Una sequenza di diagrammi per la realizzazione del caso d'uso superordinato A.

## Serie di informazioni nei sistemi di sistemi interconnessi

Un'area in cui la maggior parte delle organizzazioni impiegano molti sforzo è nel comprendere come gestire gli artefatti e le dipendenze. Abbiamo parlato precedentemente delle dipendenze tra i modelli del caso d'uso subordinato e modelli di progettazione. Esistono anche delle questioni generali di dipendenze da considerare. Quando un sistema attraversa un passaggio del ciclo di vita, verranno prodotti artefatti che possono essere organizzati in serie di informazioni [8], vedere la figura 14. Tali serie sono organizzate in base agli artefatti che si evolvono "insieme".

- È possibile personalizzare i contesti di ogni serie a seconda del tipo di applicazioni che si sta costruendo, ma le serie restano immutate.
- Bisogna capire le dipendenze tra i gruppi, così da mantenere una tracciabilità tra gli artefatti in modo efficace.

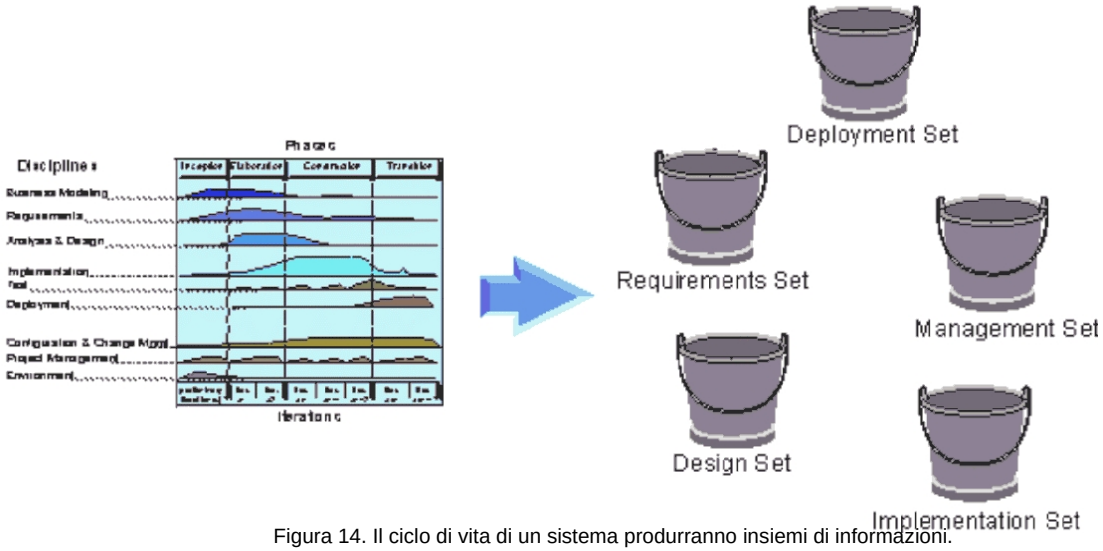


Figura 14. Il ciclo di vita di un sistema produrranno insiemi di informazioni.

In un sistema di sistemi interconnessi, i sistemi superordinati e subordinati produrranno i propri insiemi di informazioni, vedere la figura 15.

- Un insieme di informazioni subordinate ha una dipendenza dal corrispettivo insieme di informazioni superordinato.
- Il tipo di contenuto può variare in insiemi di informazioni corrispondenti tra sistemi subordinati in quanto i tipi di applicazione differiscono.
- L'insieme di informazioni subordinate corrispondenti dovrebbero essere indipendenti, a meno che soddisfino le stesse interfacce di sottosistemi definite nel sistema superordinato.

L'impegno investito nel sostenere la tracciabilità tra gli artefatti nel sistema superordinato dovrebbe essere minimo. Si dovrebbe dare priorità alla gestione della tracciabilità nel sistema.

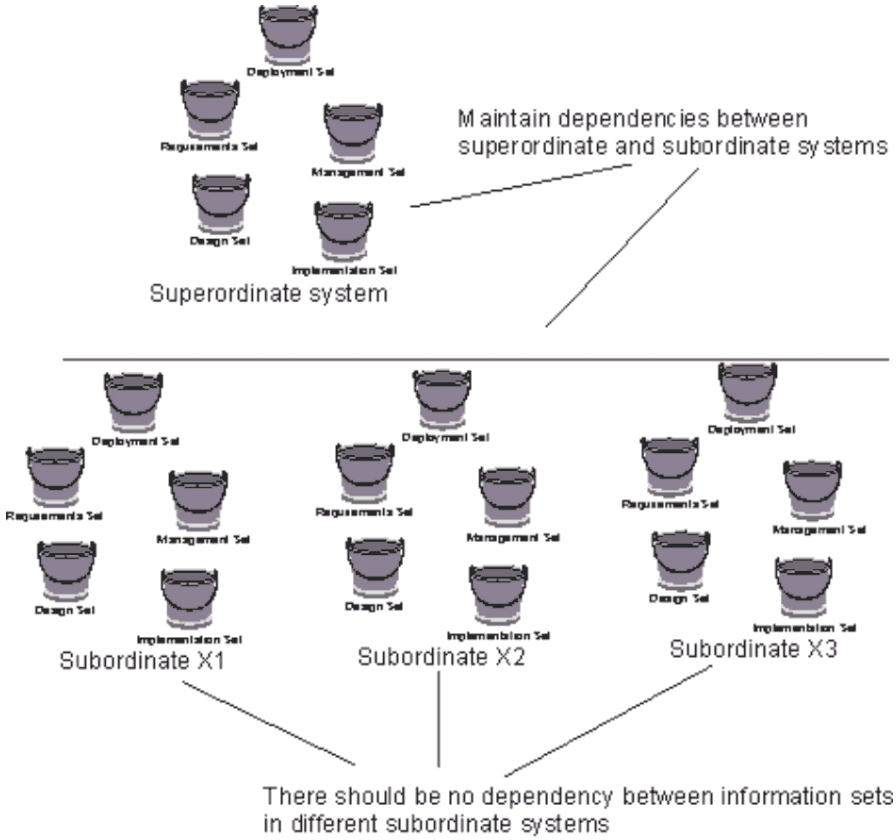


Figura 15. Ogni sistema in un sistema di sistemi interconnessi produrrà il proprio insieme di informazioni.

## Architettura nei sistemi di sistemi interconnessi

Ogni sistema nel sistema di sistemi interconnessi, sia superordinati che subordinati, dovrebbe avere definita la propria architettura.

Per il sistema superordinato, un documento di architettura dovrebbe descrivere:

- Casi d'uso o scenari principali del sistema superordinato.
- Strutturazione a livelli del sistema di sistemi interconnessi.
- In che modo gestire il riutilizzo in sistemi subordinati e cosa è possibile riutilizzare.
- I meccanismi chiave e relativa implementazione, i più generici abbastanza per essere utilizzati da tutti i sistemi subordinati. Ad esempio, tutti i sistemi subordinati dovrebbero utilizzare meccanismi comuni per la comunicazione, segnalazione degli errori, e gestione dell'errore, altrimenti il sistema superordinato non funzionerà in modo omogeneo.

Per il sistema subordinato, un documento di architettura dovrebbe specificare:

- Il ruolo del sistema subordinato all'interno del sistema di sistemi interconnessi.
- Casi d'uso o scenari principali del sistema subordinato.
- In che modo il sistema subordinato utilizzerà la struttura a livelli definita per il sistema di sistemi interconnessi. In altre parole è necessario definire come il sistema subordinato adempirà al ruolo definito nell'architettura a livelli del sistema superordinato.

- Quali meccanismi chiave generici saranno utilizzati e in che modo e quali meccanismi specifici all'applicazione vengono aggiunti.
- Come verrà applicato il riutilizzo. In modo specifico, quali sottosistemi sono comuni tra due o più sistemi subordinati e quali meccanismi vengono costruiti per consentire ai sistemi subordinati di comunicare.

## Relazioni tra sistemi

Si è potuto vedere che le consuete attività di sviluppo sistema possono essere applicate anche ai sistemi implementati da sistemi di sistemi interconnessi. Ciò costituisce un vantaggio in quanto significa che non è necessario gestire tali sistemi in modo considerevolmente diverso da quello utilizzato per altri sistemi. Si ottiene inoltre una positiva separazione del sistema superordinato dalla propria implementazione nella forma di altri sistemi subordinati. Ogni sistema in un sistema di sistemi interconnessi ha un proprio ciclo di vita. Poiché ogni sistema può avere caratteristiche differenti, è possibile effettuare alcune variazioni nel processo di sviluppo per produrli. Nei termini del Rational Unified Process [2], si avrà un diverso caso di sviluppo per ogni sistema.
Un'ultima nota sull'indipendenza dei sistemi coinvolti in un sistema di sistemi interconnessi:
dare prima uno sguardo ai sistemi subordinati. Ogni sistema implementa un sottosistema nel modello di progettazione del sistema superordinato. I sottosistemi dipendono l'uno dall'interfaccia dell'altro e non esplicitamente l'uno dall'altro, vedere la figura 12. È dunque possibile cambiare un sottosistema con una nuova versione senza danneggiare altri sottosistemi, nella misura in cui il nuovo sottosistema è conforme alla stessa interfaccia. La stessa relazione occorre tra i sistemi subordinati, ognuno dei quali vede ciò che lo circonda come un insieme di interfacce. Ciò significa che è possibile scambiare un sistema con un altro, purché il nuovo sistema svolga gli stessi ruoli verso altri sistemi, ad esempio se può essere rappresentato dalla stessa interfaccia. I sistemi fanno riferimento alle relative interfacce come specificato nelle relazioni corrispondenti tra sottosistemi ed interfacce nei sistemi superordinati.

Nel modello di caso d'uso di un sistema subordinato, le interfacce degli altri sistemi subordinati con cui interagisce vengono rappresentate come attori. Si può dire che un sistema subordinato consideri le interfacce di un altro sistema come offerte dagli attori corrispondenti, di conseguenza non deve mai fare riferimento diretto all'altro sistema, vedere la figura 16. Notare che l'interfaccia B ricorre in vari posti nella figura 12, indicando che si tratta realmente della stessa interfaccia a cui si riferiscono i sottosistemi nel sistema superordinato e i sistemi subordinati corrispondenti.

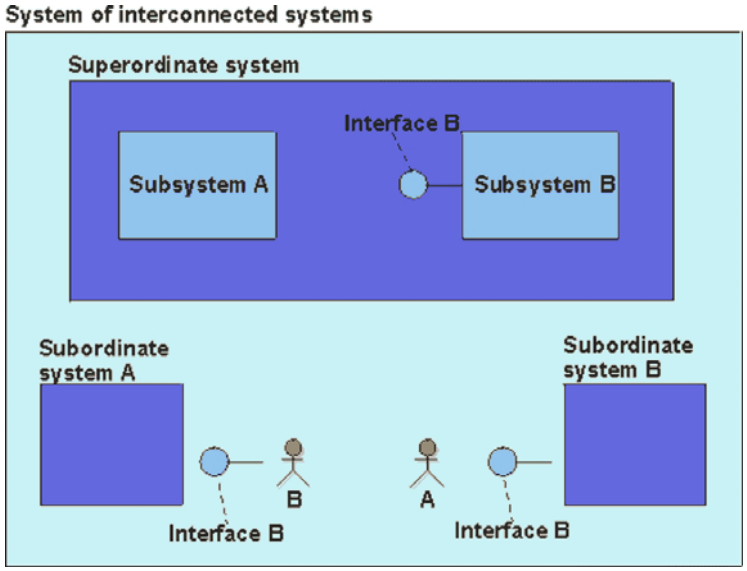


Figura 16. I sottosistemi del sistema superordinato dipendono l'uno dall'altro solo tramite le proprie interfacce. I sistemi subordinati in fase di implementazione dunque hanno lo stesso tipo di dipendenza. Nel modello del sistema superordinato, il sottosistema B fornisce l'interfaccia B ad altri sottosistemi. Il sistema subordinato corrispondente B dunque deve fornire la stessa interfaccia B ad altri sistemi subordinati.

Vediamo cosa succede al sistema superordinato e qual è la relazione con i sistemi subordinati. È indipendente dai propri sistemi di implementazione nel seguente modo: ognuno di tali sistemi è solo un'implementazione di ciò che è stato specificato nei modelli del sistema superordinato, non è parte della specifica. Per motivi pratici è necessario definire dei link di tracciabilità tra sistemi a diversi livelli, al fine di tracciare i requisiti. Il modo migliore per farlo è definire tali link solo tra le interfacce, vedere la figura 11. Infatti, si potrebbe addirittura dire che i sistemi subordinati non siano nient'altro che implementazioni che forniscono le interfacce definite nei modelli superordinati.

Ma questo vale per i sistemi che fungono da esempi non sufficienti. Un'interfaccia non specifica niente di più di ciò che avviene ad un punto di interazione preciso. Un sistema subordinato può avere centinaia di interfacce, ed ogni interfaccia decine di operazioni. Non è consigliabile collegare un input di una interfaccia ad uno o più output di altra interfaccia in una descrizione di interfaccia. Per questo motivo sono necessari casi d'uso per spiegare le semantiche del sistema subordinato.
Si può concludere che ogni sistema coinvolto nell'implementazione di un sistema da parte di un sistema di sistemi interconnessi è indipendente dagli altri sistemi, ma questi dipendono fortemente dalle reciproche interfacce. Questo fornisce un buona piattaforma per uno sviluppo parallelo dei sistemi subordinati.

## Aree di applicazione

Le tecniche di architettura e modellazione per i sistema di sistemi interconnessi possono essere utilizzate per vari tipi di sistemi, quali:

- sistemi distribuiti
- sistemi molto ampi o complessi
- sistemi che combinano diverse aree di business
- sistemi che riutilizzano altri sistemi
- sviluppo distribuito di un sistema

La situazione può anche essere capovolta: da un insieme di sistemi già esistenti, viene definito un sistema di sistemi interconnessi assemblando i sistemi. In alcuni casi infatti, i sistemi grandi si evolvono in questo modo nelle fasi iniziali dello sviluppo. Si comprende di avere dei sistemi che possono essere interconnessi e si crea così un "sistema vasto" che aggiunge valore più che due sistemi separati. Per alcuni sistemi in cui è possibile vedere diverse parti del sistema come sistemi a se stanti, è consigliabile definirlo come un sistema di sistemi interconnessi. Se per ora questo è un sistema singolo, potrebbe risultare necessario in seguito dividere il sistema in diversi prodotti separati, a causa di uno sviluppo distribuito, motivi di riutilizzo o esigenze del cliente di acquistare una sola parte, per citare alcuni esempi.

In conclusione daremo un sguardo approfondito ad una coppia di casi in cui può essere utilizzata l'architettura per sistemi di sistemi interconnessi. Per ogni esempio, mostreremo che il sistema in questione dev'essere considerato sia come un sistema singolo che come un gruppo di sistemi separati, ad indicare che dovrebbe essere trattato come un sistema superordinato implementato da un sistema di sistemi interconnessi.

### Sistemi su vasta scala

La rete telefonica è probabilmente il più vasto sistema di sistemi interconnessi del mondo e costituisce un ottimo esempio in cui sono necessari più di due livelli per gestire la complessità. È anche un esempio di un caso in cui il sistema superordinato ad alto livello è gestito da un organismo di standardizzazione e diverse compagnie competenti sviluppano uno o più sistemi subordinati che devono conformarsi a questi a tale standard. Parleremo della rete di telefonia mobile GSM (Global System of Mobile Telephony) per dimostrare i vantaggi dell'implementazione di un sistema su vasta scala come sistema di sistemi interconnessi.

La funzionalità di un sistema così vasto unisce diverse aree di business. Ad esempio, lo standard GSM copre l'intero sistema, dall'abbonato che chiama a quello che riceve la chiamata. In altre parole, include entrambi i comportamenti dei telefoni mobili e i nodi della rete. Poiché diverse parti del sistema sono prodotti propri acquistati separatamente, anche da diversi tipi di clienti, dovrebbero essere trattati come sistemi autonomi. Ad esempio, una compagnia che sviluppa sistemi GSM completi, venderà i telefoni mobili agli abbonati e i nodi della rete agli operatori telefonici. Questo è un motivo per cui trattare diverse parti di un sistema GSM come sistemi subordinati differenti. Un altro motivo è che ci vorrebbe troppo tempo per sviluppare un sistema così vasto e complesso come GSM come se fosse un singolo sistema, le diverse parti dovrebbero essere sviluppate in parallelo con vari team di sviluppo.

D'altra parte, poiché lo standard GSM copre l'intero sistema, si ha motivo di considerare il sistema anche come unitario, cioè un sistema superordinato. ciò aiuterà gli sviluppatori a capire il dominio del problema e come le diverse parti sono collegate tra loro.

### Sistemi distribuiti

Per i sistemi distribuiti su diversi sistemi di computer, l'architettura per i sistemi di sistemi interconnessi è adatta. Per definizione, un sistema distribuito consiste sempre di almeno due parti. Poiché le interfacce ben definite sono necessarie nei sistemi distribuiti, tali sistemi sono adatte anche per essere sviluppate in modo distribuito, cioè da team di sviluppo autonomi che lavorano in parallelo. I sistemi subordinati di un sistema distribuito possono essere venduti come prodotti autonomi. È naturale dunque considerare un sistema distribuito come un gruppo di sistemi separati. I requisiti per un sistema distribuito copre in genere la funzionalità dell'intero sistema e a volte le interfacce tra le diverse parti non sono predefinite. Inoltre, se il dominio del problema è nuovo agli sviluppatori, è necessario che questi prima considerino la funzionalità dell'intero sistema, senza tener conto di come sarà distribuito. Questi sono due motivi importanti per considerarlo come un sistema singolo.

### Riutilizzo di sistemi precedenti

Molto spesso i sistemi vasti riutilizzano sistemi precedenti, che possono essere descritti come sistemi subordinati. Si dovrà dunque "riprogettare" un modello di caso d'uso e forse di analisi per il sistema precedente, al fine di comprendere come funziona nel contesto allargato del sistema superordinato. Tali modelli "riprogettati" non devono essere necessariamente completi, come minimo devono coprire la funzionalità del sistema precedente che ha impatto diretto con la funzionalità del resto dei sistemi di sistemi interconnessi o che necessitano modifiche.

### Utilizzo di pacchetti prefabbricati

Un sistema può essere un'integrazione e personalizzazione di due o più pacchetti prefabbricati. Un buon esempio è il sistema Enterprise Resource Planning (ERP). Molti sistemi ERP sono una composizione di sistemi subordinati come MRP (Material Resource Planning), Inventory Management, Supply Chain Management ecc. Simili composizioni sono disponibili in altre aree come risorse umane o applicazioni di buste paga. Sono come sistemi prefabbricati da specializzare e interconnettere con altri pacchetti standard per ottenere un sistema completo. Per capire cosa fanno i gruppi di pacchetti insieme è necessario il sistema superordinato. Questo è il caso che molti clienti della comunità finanziaria affrontano oggi.

## Riepilogo

Questo documento introduce un pattern strutturale per sistemi composti da sistemi interconnessi. Questo costrutto consente la ricorsività non solo in un modello, ma considera anche ciascun sottosistema come un sistema a se stante e la ricorsività avviene tra tutti gli insiemi di artefatti di ciascun sottosistema. L'architettura introdotta viene utilizzata per sistemi che sono implementati da sistemi di comunicazione diversi.

Ciascun sistema coinvolto viene descritto dal proprio insieme di modelli, indipendenti dagli altri modelli di sistemi.

I vantaggi dell'utilizzare tale tecnica sono ovvi: è possibile avere un approccio con problemi alquanto complessi e capirli utilizzando una tecnica di divisione e acquisizione. Tuttavia, il lato negativo è che si rischiano pianificazioni più eccessive e desincronizzate. Si sono visti anche esempi che le organizzazioni ritengono molto difficile impiegare un ciclo di vita iterativo al sistema superordinato, correndo il rischio di rimandare i rischi alla fine del ciclo di vita del sistema superordinato. È anche necessario far si che venga seguita una strategia di riutilizzo efficace e comprensiva, per evitare lo sviluppo di un gruppo di sistemi inutili.

Gli esempi offerti in questo documento illustrano che l'architettura per i sistemi di modellazione di sistemi interconnessi sono utilizzabili in molte aree di applicazione diverse. È infatti possibile utilizzare l'architettura suggerita in qualsiasi sistema dove è possibile visualizzare le diverse parti come sistemi a se stanti.

## Riferimenti

[1] Jacobson, I.; Palmkvist, K. e Dyrhage, S., Systems of Interconnected Systems, ROAD, 2(1), 1995. [2] Rational Unified Process versione 5.1.

[3] Rumbaugh, J.; Booch, G.; Jacobson,I., UML Reference Manual, Addison Wesley Longman, 1999. [4] Herbert A. Simon, The Sciences of the Artificial, MIT Press, 1981.

[5] Jacobson, I.; Bylund, S.; Jonsson, P., Using Contracts and Use Cases to Build Plugable Architectures, Journal of Object-Oriented Programming, maggio/giugno, 1995.

[6] Jacobson, J.; Griss, M.; Jonsson, P., Software Reuse - Architecture, Process and Organization for Business Success, Addison Wesley Longman, 1997.

[7] Jacobson, I., Use Cases in Large-Scale Systems, ROAD, 1(6), 1995.

Rational®
the software development company

Sedi principali:

Rational Software
18880 Homestead Road
Cupertino, CA 95014
Tel: (408) 863-9900

Rational Software
20 Maguire Road
Lexington, MA 02421
Tel: (781) 676-2400

Numero verde: (800) 728-1212
E-mail: info@rational.com
Sito Web: www.rational.com
Sito internazionale: www.rational.com/worldwide