

## Lezione 8 modulo 5

In questo modulo riprendiamo i concetti legati all'Application Layer di Archimate e li applicheremo a quegli esempi che sono già stati introdotti e discussi per quanto riguarda il business layer. Quindi, vedremo alcuni esempi di applicazione dell'application layer. Giusto per rinfrescarci la memoria, riprendiamo l'esempio Speedy; l'esempio Speedy dove abbiamo visto nella sua rappresentazione con il business layer, diciamo in formato compatto, dove non avevamo ancora dettagliato quelli che erano i passi all'interno del processo, era un esempio in cui supponevamo che il top manager poteva utilizzare un servizio, grazie al quale poteva crearsi un proprio report. Era un servizio che non poteva essere utilizzato in maniera self service, quindi non utilizzando un'applicazione web a disposizione, ma faceva riferimento all'utilizzo di un data analyst, quindi un essere umano, che forniva appunto questo servizio attraverso un servizio di helpdesk. Nella sua versione estesa è praticamente la stessa struttura, con dettagliati quelli che sono i passaggi che dovevano essere svolti appunto dal data analyst, per riuscire a fornire questo servizio. Vediamo ora come è possibile definire, attraverso un'ipotesi di quello che può essere il portafoglio applicativo utilizzato per realizzare questo tipo di servizio, come è possibile definire il layer applicativo. Il testo riportato qui descrive quella che è un'ipotesi di portafoglio applicativo, nello specifico per offrire il servizio di cui sopra, Speedy si appoggia a un data warehouse. In particolare, un reporting module offerto da questo data warehouse, ha l'obiettivo, ha il compito di interagire con il sistema di data mart, quindi l'elenco dei data mart che fanno riferimento, appunto, al data warehouse, che permette quindi di realizzare i famosi report che poi saranno utilizzati dal data analyst e offerti, quindi, al top management. A supporto di questo reporting module, si ipotizza che il data warehouse, periodicamente, legge i dati rispetto ai prodotti che poi saranno analizzati dal top manager, che sono gestiti però da un altro sistema, che è il sistema di ERP; sistema di ERP che comunica attraverso un API. Per realizzare l'application layer, partiamo dal pattern solito, il pattern di base e quindi andiamo a costruire quelli che sono gli elementi principali. Quindi, abbiamo un application component, abbiamo un application interface, abbiamo detto che dal punto di vista comportamentale, un component implementa, viene assegnato a una funzione e abbiamo l'application service. Cosa possiamo dire? possiamo dire che, innanzitutto, dobbiamo definire quelle che sono le relazioni che legano questi vari elementi, secondo il modello standard, appunto, del pattern; cosa che abbiamo già visto sia nel modello di business e che si ripete anche nel modello applicativo. Secondariamente, andiamo a definire quali sono questi elementi: l'application component, nel nostro caso, è il data warehouse. Il data warehouse, che offre una funzionalità di reporting; funzionalità di reporting, che permette l'implementazione del reporting service e che viene offerta, abbiamo detto, attraverso una interfaccia web. Quindi stiamo dicendo che il nostro data warehouse offre una funzione di reporting, che è esportata attraverso un servizio, che poi collegheremo a livello di business e che può essere acceduto attraverso una graphical user interface. Potremmo dire di aver finito, però, se vogliamo essere precisi, qua è specificato in maniera chiara che esiste un reporting module, che è un modulo specifico all'interno del data warehouse. Il data warehouse quindi è un contenitore di tanti moduli e, nello specifico, quello che andiamo a dire è che è andato a specificare che è corretto modellare, perché siccome è stato specificato nel testo è corretto modellare; è il reporting module. Reporting module che è parte del mio data warehouse, quindi lo inseriamo sempre col discorso di composition come abbiamo detto. Sarebbe più corretto, quindi, una volta che abbiamo definito il reporting module, fare l'assegnamento del reporting function, assegnare questa funzione direttamente a questo modulo specifico. In questo modo noi abbiamo definito praticamente tutto e ci manca solo da inserire l'elemento passivo, che qui è ben definito, che sono i cosiddetti data mart. Ecco, rispetto all'elemento passivo, così come avviene nei pattern tradizionali, dobbiamo andare a definire quella che è la relazione di accesso fra l'elemento comportamentale e l'elemento passivo. Ecco, in questo caso, quello che succede è che il reporting function, principalmente, accede in lettura su questo datamart, rispetto a quello che abbiamo visto per il business layer, è, a livello applicativo, un po' più significativo andare a definire anche qual è il verso di questa operazione, il verso di questa freccia. Mentre nel business layer, abbiamo detto, consideriamolo come un accesso, diciamo che, in qualche modo, accede quel business object, qui è un po'

più corretto andare a definire anche il verso: se accedo in lettura, in scrittura, o in lettura e scrittura, oppure non lo so, lo lascio generico e metto un accesso. Ecco, il reporting function, è più utile dire che legge il datamart, perché accede al datamart in modalità lettura. In questo modo, io sono andato a costruire, diciamo, la prima parte del mio layer applicativo, perché, come è descritto anche qui, nel testo, in realtà il mio livello applicativo si basa anche su un'altra applicazione, che è quella dell'ERP, che attraverso un servizio di sincronizzazione, di condivisione dei dati, offre alcuni dati ad altri componenti e nello specifico, questi altri componenti è il nostro reporting module. Oppure, potremmo dire, più in generale, qui è abbastanza generico, possiamo ipotizzare che sia il reporting module, oppure l'intero data warehouse. Per questo motivo è utile e all'inizio conviene essere anche forse troppo ridondanti e ripetitivi nella descrizione, andare a inserire l'intero pattern: quindi, anche qui, abbiamo un ERP che offre i propri servizi, però in questo caso attraverso un API. Non sto parlando di ERP in generale, sto parlando di quella sezione dell'ERP che adesso a me interessa. L'ERP offre un servizio applicativo, che potremmo chiamare sharing service, che è la realizzazione di una funzione che chiameremo sharing function. Sempre le solite relazioni, qui la cosa importante è andare a riprendere il fatto che questa condivisione di dati fa riferimento a un data object specifico, che è quello dei prodotti. E anche in questo caso, ha senso dire: guarda che la mia funzione legge i prodotti. È ovvio che poi l'ERP nel suo ciclo di vita andrà a leggere e scrivere i prodotti, ma in questo caso noi stiamo definendo che va a leggere, appunto, i prodotti. Come si collegano questi due elementi? Qui è un servizio, che viene usato da un altro modulo; quindi c'è una specie di integrazione, a livello applicativo, fra due elementi applicativi; tranquillamente è un servizio, che invece che fornire un servizio a livello di business, fornisce un servizio a un altro livello applicativo. Quindi noi abbiamo la possibilità, in Archimate, a livello applicativo, di andare a definire quella che è l'integrazione fra due moduli applicativi. Detto questo, per comodità ricopio tutto. È, diciamo così, anche abbastanza evidente come sia un po' troppo dettagliato questo elemento, ci sono tutto un insieme di dettagli che sono ridondanti e creano più che altro rumore. Vediamo come è possibile, utilizzando anche le relazioni derivate, andare a fornire una rappresentazione più compatta che comunque non inficia la rappresentazione del mondo che stiamo appunto descrivendo. Innanzitutto, come possiamo vedere, un ERP viene assegnato a una sharing function, che realizza uno sharing service e qui ho un percorso che è di tipo strutturale: strutturale questo, strutturale questo, oppure questo, strutturale questo, strutturale quest'altro...Significa che secondo questo percorso, io posso andare a rappresentare comunque un collegamento diretto fra ERP e sharing service con quella che è la sua relazione più debole fra queste due, in questo caso la realization; o più debole fra queste due, in questo caso l'assignment. Siccome il mio obiettivo legare queste due direttamente, la più debole fra assignment e realization è solo la realization, quindi io potrei anche, banalmente, fare questo, eliminare questo, eliminare quest'altro.. Abbiám perso un pezzo: il product non è più collegato a nulla, però, se vi ricordate, l'ERP era assegnato alla funzione, che accedeva al product. L'accesso, quale relazione, è una relazione di dipendenza e sappiamo che quando abbiamo un path che connette due elementi in maniera indiretta, composto da elementi strutturali ed elementi di dipendenza, possiamo sostituirlo direttamente con la relazione di dipendenza, mantenendo ovviamente quello che era il verso che avevamo definito prima. Possiamo anche rappresentarlo in questo modo. Quindi, in maniera del tutto compatta io sto dicendo: guardate che l'ERP offre questo servizio e questo servizio serve il reporting module. Oppure posso essere ancora più compatto, senza perdere però significatività. In questo caso io sto dicendo che un ERP ha una relazione strutturale col service, che ha una relazione comportamentale con il reporting module; posso anche sostituirla direttamente con una relazione serve, che è la relazione di dipendenza, con reporting module e eliminare lo sharing service. In questo modo io sto dicendo che l'ERP offre un servizio al reporting module. Qui non lo sto dettagliando, se metto a disposizione direttamente questo elemento, non sto dettagliando esattamente cosa c'è dietro le quinte, però chi conosce Archimate, può, andando a riprendere quello che è il pattern di base, scoprire che dietro, in realtà, c'è il concetto di funzione, c'è il concetto di interfaccia, se voglio esplicitare l'interfaccia e dire che ho un API la posso mettere, anche se in realtà è abbastanza intuitivo il fatto che servire un modulo attraverso un altro modulo, implica necessariamente un'interfaccia di tipo applicativo. Se voglio ancora essere più preciso, posso ulteriormente arricchire questo



modello, andando a indicare non solo la dipendenza servizio-servito, quindi client e server, ma anche a definire quella che può essere il flusso informativo fra i due, il flusso informativo rilevante fra i due; quindi utilizzando una relazione di tipo Flow, io vado a indicare in maniera chiara il fatto che i dati fluiscono dall'ERP al reporting module. Ecco, dal punto di vista informatico sappiamo benissimo che quando un client chiama un server c'è uno scambio di informazioni. Ecco, qui a me non interessa tanto il protocollo che viene utilizzato, che richiede necessariamente una bidirezionalità, a me interessa unicamente dire qual è il flusso rilevante, caratterizzante questo tipo di servizio, che nella maggior parte dei casi segue il verso del serve, però può anche succedere che ci sia il verso opposto, quelli sono casi abbastanza particolari. Detto questo, andiamo a vedere come è possibile collegare questo livello applicativo con il livello di business. Ecco, qui non riprendo e non lo faccio a mano, vi mostro direttamente il diagramma: nella parte in basso sono andato a riprendere esattamente quello che avevamo fatto all'inizio, questo qui, quindi c'è la parte, forse riaggiustata nelle posizioni, quello che sto dicendo è tranquillamente: guardate che questo processo è sostenuto da questo servizio. Quindi il data analyst, utilizzando questa interfaccia, ha la possibilità di utilizzare i servizi offerti dal reporting service quale elemento di supporto al Customized Report Creation Process. In realtà, l'esempio che abbiamo visto l'altra volta riguardo a Speedy, aveva due accezioni: permetteva di creare il processo di offrire servizio sia in modalità manuale, quello che abbiamo appena visto, ma anche in modalità automatizzata. Infatti, se vi ricordate, abbiamo visto che era anche possibile dire, con dei passi leggermente diversi, non sto a ripeterli, però non è quello l'importante, che il top manager può utilizzare un servizio che è composto da questo processo, attraverso un sito web. Ecco, in questo caso, come posso, ipotizzando che dietro le quinte ci sia esattamente lo stesso livello applicativo? Quindi non vado a chiamare il livello applicativo, vado semplicemente a dire: guarda che questa web graphical user interface viene utilizzata dal top manager invece che essere mediata dal data analyst. Allora, in questo caso, quello che noi possiamo rappresentare, è una situazione di questo tipo: come vedete, qui vado a riprendere il business layer, qui vado a riprendere esattamente l'application layer che abbiamo appena visto, nella sua versione non compatta, però posso anche tenermi la versione compatta che abbiamo rivisto, non cambia assolutamente nulla, perché in realtà i servizi che vengono usati dal top manager, sono i servizi offerti dal reporting function. Se vi ricordate, nel caso di Self service, la web interface che viene offerta dal reporting service, o la web graphical user interface, c'è il nome leggermente diverso, non è quello importante, realizza quel website a disposizione del top manager. Il servizio è il servizio che realizza il servizio di business, quindi il top manager non fa altro che utilizzare un servizio di business che è equivalente al servizio applicativo. Quindi, tutto quello che offre questo servizio attraverso la web interface. E, le azioni che deve eseguire il top manager per utilizzare questo servizio, sono le azioni indicate da questo processo, che sono implementate dal reporting function. Quindi immagino che qui dentro ci siano tutti gli elementi che mi permettono di realizzare questo tipo di soluzione, questo tipo di processo. In questo modo, io ho fatto vedere come lo stesso livello applicativo può essere usato in due accezioni diverse, per utilizzarlo all'interno di due servizi diversi. È ovvio che questo è un caso abbastanza limite, difficilmente lo stesso livello applicativo viene usato in tanti servizi differenti. Però può anche capitare che un sottoinsieme degli elementi applicativi può essere riutilizzato in tanti ambiti. Riprendo anche l'esempio dell'insurance, l'altro esempio che abbiamo visto l'altra volta, che era caratterizzato dal fatto che il mio customer utilizzava appunto questo servizio in modalità self service, già era stato dichiarato fin dall'inizio, e una di queste attività, che dovevano essere svolte, per l'esecuzione, appunto, del processo a supporto di questo servizio, era un'attività che veniva gestita da un evaluator, quindi da una terza parte, diciamo così, comunque interna, ma non il customer e nulla di automatico, appunto, l'evaluator, supportato comunque da un sistema informativo, per la stima del prezzo. In questo caso non vado a riprendere tutti gli elementi per avere qualcosa di un po' più compatto, supponiamo che, e questa è l'ipotesi di base, supponiamo che il sistema IT della mia insurance company, è composto da un sistema CRM, che offre tutte le informazioni legate ai customer, che mi permettono di eseguire questa attività. Esecuzione di checking della reliability del mio customer, dell'affidabilità del mio customer, che può essere eseguita in maniera totalmente automatica. Poi ci sono altri due passaggi che verranno governati poi dalla

web graphical user interface, poi abbiamo la stima del prezzo, in questo caso, la stima del prezzo viene offerta grazie a un'applicazione di tipo desktop per il nostro estimator. Quindi esiste un modulo che qui è chiamato allo stesso modo del ruolo, ma è solo un caso, che offre la stima, la funzionalità di stima del bene e che serve appunto l'estimate price. Giusto come esercizio, immaginate di esplodere questi elementi e ricondurli comunque al pattern di base. Perché? Perché qui ho già offerto una visione compatta di quello che è un qualcosa di più elaborato. La stessa cosa vale anche per il CRM. Quello che ho ottenuto un po' più dettagliato, come pattern, è la parte legata all'insurance service, che invece viene gestita da un'applicazione, che è appunto l'applicazione di contract management, che offre la funzionalità di Contract Management, attraverso una web Graphical User Interface. Quindi, utilizzando ancora quella relazione doppia di realizzazione fra servizio applicativo che realizza servizio di business e la funzione applicativa che realizza il processo di business, io vado a creare quella connessione che mi permette dichiarare che questo è un processo totalmente automatizzato, almeno dal punto di vista dell'utente, che poi dietro le quinte ci sia comunque un aspetto umano, quello comunque è totalmente indipendente. In questo modulo, per semplicità, per non mettere troppi elementi, ho tolto la parte di web graphical user interface, legata al processo in generale, a questo servizio in generale, però potete, per esercizio aggiungerlo e praticamente non è altro che un'interfaccia da inserire all'interno di questa parte di business e di collegarla con un realize, con la web graphical user interface che ho definito a livello applicativo.