

Lezione 2.3

Come accennato in precedenza, Archimate è il linguaggio di modellazione per Enterprise Architecture, che copre i diversi domini di interesse appunto delle Enterprise Architecture. Noi ci concentriamo principalmente sul dominio di business, dominio organizzativo, quindi il dominio applicativo e il dominio tecnologico. In questa lezione ci concentreremo appunto sul primo dominio, sul dominio organizzativo per la realizzazione di quello che viene chiamato il Business Layer di un Enterprise Architecture. Per chi volesse approfondire e avere il dettaglio di tutti gli elementi che andranno a essere introdotti in questa lezione il consiglio è di dare anche un'occhiata a quella che è la specifica di Archimate e in particolare su quello che è il capitolo relativo al Business Layer il cui URL è indicato in questa slide. Come prima cosa, andiamo a elencare gli elementi attivi, gli elementi comportamentali e gli elementi passivi tipici all'interno di un diagramma che va a rappresentare il layer di business di un diagramma Archimate. Nello specifico, noi ci concentreremo su questi quattro elementi attivi, il Business Actor, il Business Role, il Business Interface e la Business Collaboration. Il business actor non è altro che un qualunque elemento, un qualunque attore, una qualunque entità, a livello di business, quindi può essere un customer, può essere un impiegato, può essere un manager che ha la capacità di eseguire una determinata azione, cioè di essere associato a un determinato comportamento, a un determinato behavior. Il secondo è il business role. Il business role indica più che altro una responsabilità, non è associato unicamente alla capacità di eseguire un'azione, ma è associato alla responsabilità sulla capacità di eseguire un'azione e di solito può anche accadere che lo stesso ruolo può essere ricoperto da uno o più attori. L'altro elemento è la Business Interface. La Business Interface identifica il punto di accesso a un servizio, cioè le modalità con cui l'utente del servizio può effettivamente interagire con il servizio. Vi sono diverse tipologie di questa Business Interface, qui si parla ad esempio degli elementi, negli esempi più evoluti di sito web o portale web. Si può parlare anche di un call center, si può parlare di una filiale dove c'è una persona dedicata all'erogazione di questo servizio. Poi c'è la Business Collaboration, che è un'aggregazione di due o più elementi attivi all'interno della mia organizzazione, che concorrono tutti nella realizzazione, nell'esecuzione di un determinato elemento comportamentale, quindi ogni qualvolta io ho più di un attore piuttosto che più di un business role che concorrono assieme all'esecuzione di un'attività, per esempio di un processo in maniera collaborativa allora possiamo utilizzare la Business collaboration. Vediamo ora la relazione nello specifico, che esiste fra ruoli ed attori. Come detto, abbiamo detto appunto che un ruolo può essere ricoperto da diversi attori. In questo caso, attraverso questo semplice diagramma, noi stiamo dicendo che un determinato cliente, che è un attore, sta ricoprendo il ruolo di assicurato, di insurant. Quindi vedete, che è appunto più un discorso di responsabilità di ruolo, appunto quello dell'insurant, che può essere ricoperto da un determinato attore all'interno del mio diagramma che è il customer. Lo stesso ruolo può anche essere ricoperto da più attori, quindi può anche succedere che vi siano più attori che sono assegnati, perché questo è il significato di questo tipo di relazione, a eventuali ruoli, quindi l'assegnamento è la relazione che si utilizza nel momento in cui io voglio specificare l'assegnamento, appunto, la relazione che esiste fra un attore e un ruolo. Tornando al discorso della collaborazione, in questo semplice diagramma vediamo come due ruoli, ma potrebbero essere anche due attori, perché abbiamo detto che la Collaboration si esprime come aggregazione di due o più elementi attivi, quindi in questo caso i due ruoli che sono il venditore e il compratore, definiscono una partnership, una trading partnership e questo è quello che identifica la cosiddetta Business Collaboration. La Business Collaboration, come vedete, è unita da il buyer e unita da il seller attraverso una relazione particolare che è la relazione di aggregazione. La relazione di aggregazione presenta un diamante vuoto a un'estremità e nessun elemento decorativo all'altra estremità. Importante adesso è dare qualche informazione su questa tipologia di relazione, che è strettamente collegata e molto simile con quell'altra relazione che viene chiamata di composizione, che viene rappresentata sempre da un diamante che però è colorato, quindi si presenta di colore nero. Come detto, questa è l'aggregazione, significa che esiste un elemento che aggrega e un elemento che è aggregato, quindi abbiamo il buyer e il seller che sono aggregati all'interno di un elemento aggregante che è il trading partnership. La relazione



che esiste fra seller e buyer e trading partnership, è una relazione che qualcuno potrebbe interpretare come una relazione di composizione, portando quindi a una confusione di termini rispetto all'altro tipo di relazione che abbiamo detto che è appunto la relazione di composizione che, come detto, viene rappresentata da un diamante colorato di nero, quindi pieno, invece di questo diamante vuoto. Che differenza esiste fra relazioni di aggregazione e relazioni di composizione? In entrambi i casi definisce una relazione di appartenenza, potremmo dire, fra elementi attivi e un altro elemento attivo, solo che nel caso dell'aggregazione, la relazione che esiste rispetto al ciclo di vita dei vari elementi attivi è più lasca, è più debole rispetto a quello della composizione. In altre parole, la domanda che dobbiamo porci è: se io elimino non ha più senso parlare di trading partnership, ha ancora senso parlare di seller e buyer? Sì, perché comunque nel mio modello io ho ancora qualcuno che vende e qualcuno che compra, non necessariamente magari esiste questa collaborazione. Quindi il ciclo di vita dell'elemento che aggrega è indipendente dal ciclo di vita degli elementi che sono aggregati. In altre parole, quello che viene detto è che nel momento in cui termina la vita dell'elemento che aggrega, non è detto che termini anche la vita, diciamo così, non ha ancora più senso, non ha più senso pensare a cosa sono, all'esistenza del seller e del buyer, degli elementi che sono aggregati. Ecco, questo cambia nel momento in cui noi utilizziamo l'operatore di composizione. Ecco, in questa situazione non senso, principalmente perché la collaborazione, per sua definizione, è un'aggregazione di elementi, quindi non ha senso mettere la composizione, ma supponiamo di avere degli elementi attivi che sono legati attraverso l'operatore di composizione. Vi ricordo che adesso noi facciamo degli esempi di aggregazione, composizione, assegnamento su elementi attivi, ma in realtà questo tipo di relazioni vengono usate su qualunque, in diversi, non proprio su qualunque, ma in diversi contesti, unendo diverse tipologie legate a diversi aspetti dei vari domini, quindi, quello che vediamo adesso per il livello organizzativo e il livello di business può anche essere applicato agli altri livelli. Ad ogni modo, se noi immaginiamo adesso di avere la composizione come elemento di unione fra questo elemento, trading partnership e seller e buyer, io indico una relazione molto più stretta. Sto indicando che nel momento in cui il trading partnership non ha più senso di esistere, allora non ha più senso di esistere nemmeno il seller e il buyer; questo è per sottolineare, appunto, il fatto di avere questo tipo di diverse relazioni. Mettere una o mettere l'altra significano cose diverse. Bisogna quindi iniziare a pensare che qualunque simbolo grafico si va a mettere all'interno di un diagramma, Archimate in questo caso, ma un qualunque linguaggio grafico porta con sé una semantica ben precisa e quindi quello che vado a scrivere è un qualcosa che, per chi conosce correttamente la semantica, la sintassi e il lessico di Archimate, viene recepita, viene acquisita in senso corretto anche da chi legge questo diagramma. Questo è l'unico modo che si ha per effettivamente trasferire conoscenze rispetto alla nostra Enterprise Architecture fra i diversi attori che gravitano attorno alla definizione e all'uso dell'Enterprise Architecture. Giusto per concludere il commento di questo diagramma, qui abbiamo un esempio in cui abbiamo un unico attore che ricopre, può ricoprire, sia il ruolo di buyer che il ruolo di di di seller, ovviamente non allo stesso momento, in due momenti diversi, due customer diversi possono ricoprire o un ruolo o l'altro. Esiste un altro tipo di interazione che può avvenire fra ruoli o anche fra attori diversi, che viene definito attraverso il concetto di servizio. Qui vengono utilizzati solo gli elementi attivi, poi vedremo che si può articolare in maniera un po' più comprensibile, un po' più completa, introducendo anche l'elemento comportamentale per eccellenza per descrivere il servizio che è il business object. In questo caso noi abbiamo che un seller offre una certa interfaccia, dialoga attraverso una certa interfaccia, che è il web store, che serve il buyer. Quindi, rispetto a prima, io non ho una situazione paritetica in cui il seller e il buyer, in qualche modo loro collaborano per raggiungere un determinato obiettivo, ho qualcuno che offre un determinato servizio attraverso una determinata interfaccia e che viene usata da un altro attore, in questo caso un altro ruolo. Qui sottolineo la differenza nell'utilizzo dell'operatore della relazione, in questo caso utilizziamo appunto la relazione di composizione. Per dire che cosa? Se io non ho più un seller, perché per qualche motivo non c'è più nessuno che vende, non ha più senso avere questa tipologia di interfaccia. Ecco, per uno che conosce un po' UML e legge questo tipo di diagramma, qualcosa non torna, perché noi stiamo dicendo che un'interfaccia è un elemento che compone un ruolo: c'è qualcosa che non torna, sembra quasi che una persona è composta da



interfacce, è un modo, diciamo così, che viene utilizzato all'interno di Archimate per indicare il fatto che una determinata, poi, funzionalità su cui è responsabile nell'esecuzione un determinato ruolo può essere accessibile attraverso un'interfaccia che questo ruolo in qualche modo mette a disposizione. Il venir meno dell'esistenza del ruolo rende inutile la presenza anche dell'elemento che lo compone, quindi l'interfaccia. Ultima nota a margine, abbiamo anche l'opera, la relazione di serve che indica chi offre il servizio, offre il servizio a chi viene offerto il servizio, quindi la freccia va verso chi è servito, parte da chi serve, verso chi è servito. Quindi attraverso questa interfaccia io offro un servizio a chi? Al buyer che è servito in questo caso. Rispetto a prima non ho, come nel caso della collaboration, come detto, qualcosa di paritetico ma ho qualcosa in cui è evidente qualcuno che offre e qualcuno che consuma, che usa quanto offerto dall'altro, quindi è tipico dell'aspetto a servizi. Passando agli elementi comportamentali, gli elementi comportamentali che noi vediamo più spesso nei nostri diagrammi sono questi quattro: il business process, che identifica una sequenza di attività, potremmo dire, anche se è più corretto dire una sequenza di elementi comportamentali di tipo business, quindi li troveremo tutti gialli, e tutti questi elementi concorrono assieme a un determinato obiettivo. Questo è diverso da quella che è la definizione di business function. Ci torneremo comunque su questa distinzione più avanti nella lezione di oggi. Per dire che cosa? Che la business function è invece una collezione, non è una sequenza, dove identifico un ordine in qualche modo, ma una collezione, quindi è più un gruppo, sempre di elementi comportamentali, questo raggruppamento è definito secondo alcuni criteri che possono essere l'utilizzo delle medesime risorse, l'appartenenza al medesimo dipartimento, che appartengono appunto alla mia organizzazione. Quindi è proprio il concetto di funzione aziendale, nel senso tipico che si ha appunto nella teoria delle organizzazioni. Poi abbiamo il business service che è la definizione esplicita di un comportamento, potremmo dire di una funzionalità senza però andare a toccare il concetto di funzione, che è una cosa diversa, una funzionalità che viene offerta all'esterno rispetto alla mia organizzazione. Poi c'è il concetto di evento, che identifica un cambiamento di stato all'interno della mia organizzazione, dovuta al accadimento appunto di un qualcosa. Un qualcosa può essere un timer che scade, un qualcosa che può essere l'arrivo di un ordine o qualunque altro elemento che fa cambiare lo stato, appunto, della mia organizzazione. Passiamo quindi agli elementi passivi. Con gli elementi passivi noi andiamo a identificare principalmente i business object. Avremo a che fare con i business object, che non sono altro che dei concetti rilevanti per la nostra organizzazione, che poi andranno a essere mappati con delle entità all'interno di database con delle tabelle. Però a questo livello io sto parlando unicamente di concetti. Quindi volendo legarlo al mondo dei database siamo, diciamo così, a quei concetti a livello concettuale, che possono essere rappresentati secondo alcuni formati, ma non li utilizzeremo. Molto spesso questo tipo di notazione, gli unici business object che sono una specializzazione appunto del business object classico, sono i cosiddetti contratti, in cui io vado a definire in maniera formale quello che è un agreement, un accordo fra alcune parti ed è molto utile nel momento in cui si definiscono i servizi. La loro rappresentazione è esattamente identica a quella del business object, però con una banda in più nella parte bassa appunto del rettangolo. Infine c'è un prodotto che è un aggregatore di servizi, di elementi passivi, che in qualche modo sono offerti a un customer, quindi un servizio con tutti gli elementi passivi che sono offerti al customer possono essere rappresentati come prodotti. Anche in questo caso non li utilizzeremo per semplicità, perché non vogliamo approfondire troppo la parte organizzativa di Archimate. Per concludere questa prima carrellata di elementi, parliamo di relazioni. È utile sottolineare il fatto che adesso spieghiamo le relazioni all'interno del livello di business. Però il significato di queste relazioni è identico a qualunque livello, su qualunque parte del mio diagramma. Io ho sempre questo tipo di relazioni con questo significato e con queste notazioni. Allora, come vedrete, vedete, abbiamo relazioni che sono classificate secondo relazioni di tipo strutturale, relazioni di dipendenza, relazioni dinamiche e tutte le altre tipologie di relazioni. Nella relazione di tipo strutturale abbiamo la composizione, l'aggregazione, la cui differenza l'abbiamo già descritta precedentemente, l'assegnamento che abbiamo già introdotto, che esprime appunto quindi l'allocation di responsabilità nell'esecuzione di un determinato behavior, la realizzazione, che indica il fatto che un elemento realizza, quindi implementa, rende concreto un altro elemento. Lo vedremo nella



relazione che sussisterà poi fra processo e servizio. Abbiamo delle relazioni di dipendenza, che identificano la offerta di un servizio, cioè il fatto che qualcuno offra un servizio e qualcun altro fruisce di questo servizio e quel serve, che abbiamo detto, nota a margine, nelle vecchie versioni di Archimate, quindi, se avrete a che fare con qualche vecchio diagramma realizzato nella versione 2.0, il verso del served era il contrario, perché non non esisteva il Serving ma esisteva lo used to. Per questo motivo la freccia andava da chi usava il servizio verso chi forniva il servizio. Ricordiamoci che noi utilizzeremo la 3.0. e una delle differenze è appunto il fatto che abbiamo unicamente questo tipo di relazione che parte da chi offre il servizio a chi usa il servizio. Poi la dipendenza può esprimere una relazione di accesso e abbiamo visto la relazione di accesso fra gli elementi comportamentali e gli elementi passivi, la descriveremo più in dettaglio, per il momento accontentiamoci che devo andare a definire, ci sono le diverse varianti, quell'assenza, nessun tipo di terminale in cui non è chiaro chi accede a che cosa, in che modalità, se è un accesso bidirezionale o un accesso monodirezionale. Soprattutto a livello di business, noi avremo unicamente un accesso monodirezionale per indicare che appunto l'elemento comportamentale accede a un oggetto passivo, può essere lettura, scrittura, cancellazione, quello che vogliamo, non ha senso adesso specificarlo e quindi in maniera del tutto generica, diciamo che vi accede. Poi c'è una relazione di influenza che non vedremo perché non fa parte, non viene utilizzato nei tre domini che interessano a noi, quello organizzativo, quello applicativo e quello tecnologico. Poi vedremo che è possibile definire anche una relazione di tipo triggering per indicare il fatto che alcune attività, l'esecuzione di alcune attività o l'accadimento di certi eventi triggerano, quindi scatenano l'esecuzione di altre attività oppure di altri eventi, oppure per indicare quello che è il flusso informativo, il flusso dei servizi fra un elemento e un altro elemento. Poi abbiamo altri tipi di relazione, la classica specializzazione che abbiamo, per esempio anche nei database, l'associazione che invece è una generica relazione per dire, guardate che questi due oggetti sono associati fra di loro senza una semantica ben precisa. Come vedete, queste relazioni sono state inserite in un determinato ordine che ha un significato ben preciso, perché identificano anche la forza della relazione. In questa slide, infatti, vediamo come quelle relazioni che abbiamo visto prima, possono essere messe in ordine dalla più forte alla meno forte. Perché è importante sapere la forza di una relazione? Perché quando andremo a definire le relazioni derivate, conoscerne la forza ci permette, partendo da una catena di relazioni, di andare a trovare, chiamiamole delle shortcut, delle scorciatoie per identificare con un'unica relazione una catena di relazioni e il conoscere quale è più forte e quale è più debole è importante. Giusto per fare un esempio, il concetto di forza e debolezza è già stato introdotto quando abbiamo parlato di composizione e aggregazione. La composizione è la più forte in assoluto perché, abbiamo detto, che gli elementi che lega in qualche modo sono legati dal punto di vista addirittura del loro ciclo di vita. Che è un po' più lasco quando vado a parlare di aggregazione. Nel caso di assignment è ancora meno lasco, perché non non parlo di un'unione di elementi, come nel caso di composizione e aggregazione, ma parlo semplicemente del fatto che ho due elementi che in un determinato momento sono in relazione per il fatto che uno è assegnato all'esecuzione di un altro elemento. Così via via per tutti gli altri, fino ad arrivare a quella più debole che definisce una relazione generica senza una semantica ben precisa. Fatta la panoramica degli elementi che caratterizzano principalmente un diagramma Archimate, soprattutto a livello business, vediamo come ha senso mettere assieme questi vari elementi. Ecco, esiste una guida che è il basic business pattern. Adesso lo vediamo a livello business, però ci sarà un basic application pattern, un basic technology pattern in qualche modo, che ci aiutano a cominciare almeno a costruire il nostro diagramma. A livello business, questo diagramma viene costruito appunto in questo modo. Innanzitutto, siccome io sto definendo un servizio offerto a una determinata clientela, è necessario identificare qual è questa clientela. È necessario definire qual è questo servizio, ovviamente utilizzerò la relazione Serve per indicare che questo servizio è offerto a questa clientela. Poi qui ho messo la clientela come attore, potrebbe anche essere un ruolo, perché abbiamo visto che esiste una certa similarità fra questi due elementi. Poi vado a dire che un processo realizza questo servizio, attraverso appunto la relazione di Realization. Un processo che adesso è indicato semplicemente con un'icona, può essere più o meno articolato, per il momento accontentiamoci del fatto di avere il concetto di processo. Sul lato destro abbiamo gli elementi attivi che sono responsabili degli



elementi comportamentali, che rappresentano un punto, il servizio. e l'implementazione di questo servizio, cosa rende possibile la fornitura di questo servizio, che sono la business interface, quindi l'interfaccia di business che è in carico di offrire, di rendere visibile questo servizio. Quindi, il customer interagisce con questo servizio attraverso questa interfaccia. Abbiamo quindi un collegamento indiretto fra la business interface e il customer, per indicare comunque che il customer può accedere a questo business interface attraverso questo business service, attraverso questa interfaccia di business. Allo stesso modo, identifico il business role come quell'elemento attivo che è reso visibile, diciamo, come abbiamo visto già in precedenza, attraverso una determinata interfaccia e che ha la responsabilità nell'esecuzione di questo processo. Quindi, in qualche modo questo ruolo è il ruolo con cui interagisce il mio customer attraverso questa interfaccia e è il ruolo che ha il compito di seguire il customer nell'esecuzione del processo. Infine, noi abbiamo che il processo accede a uno o più, in questo caso a uno, determinato business object. Vediamo adesso di calare il pattern che abbiamo appena visto in un esempio potremmo dire pratico, abbastanza intuitivo. Stiamo parlando di una clinica di un ospedale che vuole modellare il nuovo servizio offerto ai propri pazienti di prenotazione di esami. In questo caso, io con questo diagramma sto dicendo che il servizio viene offerto attraverso un call center, ma non perché qua c'è scritto Call Center Service, ma per il fatto che io ho un'interfaccia call center che è assegnata a questo servizio. Questo servizio è offerto a chi? Non a tutti, solo ai pazienti. Questo servizio è realizzato attraverso un determinato processo che adesso è generico. E poi c'è un call center operator che è il responsabile nell'esecuzione di questo processo. Il paziente, quindi, avrà a che fare tramite il call center con un call center operator. Non andrà a interagire direttamente con il mio processo, ma sarà mediato, questo accesso al processo, dall'esecuzione, dalla presenza del mio operatore. Nell'esecuzione del mio processo, infine, io andrò a lavorare con i dati legati agli esami, che qui sono rappresentati in maniera generica. Prima di andare avanti è utile andare a riprendere il discorso della layer view, che è stato presentato nella lezione precedente di introduzione su Archimate. Noi adesso siamo posizionati a questo livello perché abbiamo detto che il Business Layer si trova sopra l'application layer che si trova sopra il technology layer. Ecco, il Business Layer abbiamo detto che è anche l'unico che offre un servizio all'utente finale, che è quello che stiamo appunto facendo noi. Vedete che questo elemento non è altro che questo elemento. Abbiamo anche detto che questo vale per il Business Layer, ma anche per gli altri, è sempre utile distinguere l'interfaccia dall'implementazione di un servizio, nello specifico, le interfacce di implementazione sono divise da questa visibility line immaginaria, che avevamo già identificato anche qui per ogni elemento. Vedete che qui abbiamo la realization che ritroviamo esattamente anche qui, quindi il mio pattern non fa altro che descrivere quella che è l'interfaccia da quella che è l'implementazione e legarla secondo la struttura che abbiamo descritto prima. Quindi il mio customer, infatti, abbiamo detto che non ha visibilità su quello che accade sotto, sul mio processo, ha solo visibilità sul fatto che accede a una determinata interfaccia e dietro a questa interfaccia c'è un determinato ruolo. Nel caso del call center, il customer non conosce l'operatore, ma conosce il concetto, interagisce col concetto di call center, dietro al quale c'è un operatore generico, che però dopo nell'esecuzione è un operatore fisico, un operatore ben preciso, ma il customer non ha visibilità su quale sarà l'operatore fino a che quella esecuzione del processo non avviene. E non andrà mai a interagire con il processo vero e proprio, ma l'esecuzione sarà mediata appunto della presenza di questo ruolo. Poi vedremo che ci sarà un caso particolare in cui noi andremo a lavorare direttamente su processi automatici, soprattutto quelli che vengono offerti via web. Ecco, questi processi automatici vedrete che vanno a cambiare un minimo la struttura del nostro Business Pattern. Giusto un dettaglio su quello che avviene sopra la visibility line, ricordiamoci che, appunto, come detto, l'utente vede il servizio accessibile solo attraverso l'interfaccia che viene identificata, può anche accadere che non vi sia un'unica interfaccia ma che vi siano più interfacce. Io potrei dire, per esempio, che questo servizio, lasciamo perdere che l'ho chiamato call center, come detto è solo una label, avrei potuto chiamarlo New Examination, Booking Service generico, in questo caso è offerto sia attraverso una presenza fisica, sia attraverso un call center. Ricordiamoci però che il linguaggio Archimate prevede che vi sia un unico assegnamento fra l'interfaccia e il servizio, quindi io potrei, anche se sbaglierei, unire attraverso l'assignment direttamente desk con il servizio, così come call center con il



servizio: questo non è possibile farlo. Però esiste, diciamo così, un trucco, non è un trucco ma è un elemento che è l'elemento di junction, che non è una relazione vera e propria, ma è uno strumento che mi permette di unire più elementi, più relazioni dello stesso tipo, quindi non posso avere relazioni diverse in ingresso e relazioni diverse in uscita, ricordandosi sempre che il junction può essere usato per avere N relazioni in ingresso e una in uscita, oppure una relazione in ingresso e N in uscita. Non è possibile avere N relazioni in ingresso con N relazioni in uscita. Detto questo, io semplicemente sto dicendo che il desk e il call center, assieme, perché questa junction è di tipo and, come faccio a sapere che è di tipo and? Semplicemente perché il pallino è pieno. Se il pallino fosse vuoto e fosse bianco sarebbe un or, inteso però come xor. Ecco qui diciamo che chi ha proposto Archimate, non è stato proprio rigoroso e intuitivo nella definizione dei vari elementi, quindi questo è un and, ma ci torneremo, questo è un and. In altre parole, sto dicendo che questo servizio è offerto sia da questo che da quest'altro. Posso usarli tutti e due in contemporanea, posso usarne uno o l'altro allo stesso momento, quello lo possiamo descrivere ulteriormente andando a vedere le altre parti del mio diagramma. Altro dettaglio, invece, che riguarda quello che succede dietro la visibility line, dobbiamo appunto, dire chi è responsabile nell'esecuzione dell'elemento comportamentale che concorre alla realizzazione del mio servizio. E qui occorre fare qualche precisazione. La prima precisazione è che può esserci al più un ruolo assegnato a un processo. Al più cosa significa? Significa che posso usare, come prima, nel momento in cui può accadere che vi siano più ruoli che concorrono assieme nella realizzazione di un processo, posso usare le junction, ma anche le collaboration, in questo caso, per definire appunto questa collaborazione. Può anche succedere che non vi sia nessun assegnamento fra l'elemento, il ruolo e il processo, questo indica il fatto che il mio processo è da considerarsi come completamente automatizzato. Quindi non c'è una figura umana, potremmo dire, che accompagna l'esecuzione del processo, ma è tutto in carico all'utente finale che, attraverso uno strumento informatico, un sito web, può eseguire il processo e questo richiede una modellazione leggermente diversa e la approfondiremo fra poco. Giusto per fare un esempio, cosa succede quando ho più di un responsabile? In questo caso ho il mio business process, ho due ruoli che concorrono all'esecuzione di questo business process, posso usare il junction come abbiamo fatto prima. In questo specifico caso non ha molto senso lavorare sull'interfaccia, ma ha senso lavorare sui ruoli. Posso definire una collaborazione fra questi due ruoli e la collaborazione è assegnata al business process, perché è pur sempre un elemento attivo che viene assegnato a un elemento comportamentale. Cosa succede invece nel momento in cui ho un servizio automatico? Il pattern è praticamente lo stesso, dove però non ho nessun ruolo assegnato al mio processo. Sto dicendo che il paziente può accedere a questo servizio attraverso, in questo caso, un sito web, non ho nessuna figura umana che mi aiuti nell'esecuzione del mio servizio. Per chi è un po' più attento, avrà notato che non c'è il processo come prima, ma qui non c'entra nulla il fatto di essere automatizzato o no è solo per introdurre il nuovo concetto che è il concetto di funzione. Per dire che cosa? Che anche la funzione, così come il processo, possono essere legati attraverso una relazione di realizzazione con il servizio. Quindi, così come il processo realizzava un servizio, anche una funzione potrebbe realizzare un servizio. Non esiste differenza nelle due cose. Così come nel processo aveva un ruolo che era assegnato a un processo, posso avere un ruolo assegnato a una funzione. Ma qual'è la distinzione che esiste fra funzione e processo? Ecco, qui bisogna dare un minimo di interpretazione. E l'interpretazione che viene data in questo corso è la seguente, e non è identica per il livello di business e per il livello applicativo, perché vedremo che anche a livello applicativo avremo il concetto di funzione e processo, dove però daremo una semantica leggermente diversa e forse anche un po' più semplificata. Innanzitutto, una funzione e un processo sono, come detto, eseguite da un ruolo singolo, quindi se ho invece una commistione di ruoli, me la cavo, vuoi col junction, vuoi con la collaboration. Sia funzioni che processi, così come già detto, sono una composizione, in un caso di una sequenza, in un caso di un raggruppamento di altri elementi comportamentali. Cosa significa? Significa che un processo può essere composto da funzioni. Un processo può essere composto da processi, una funzione può essere composta da processi, una funzione può essere composta da funzioni e processi. Una funzione può essere composta da funzioni. Quindi posso, diciamo comporli un po' come voglio. Comporli cosa significa? Significa legarli fra di



loro, con il concetto, con la relazione, appunto di composizione. È importante però ricordarsi che la funzione deve essere utilizzata nel momento in cui gli elementi che io compongo hanno qualcosa in comune, il ruolo, le risorse utilizzate, il fatto di appartenere allo stesso dipartimento, ma non è mai identificato un ordine di esecuzione di questi elementi comportamentali. Nel momento in cui invece ho la necessità di definire questo ordine, allora ha più senso parlare di processo. In questo caso, gli elementi che lo compongono non sono raggruppati per l'utilizzo in comune di qualche aspetto, ma perché tutti assieme questi elementi che lo compongono, tutti assieme, puntano a raggiungere un determinato obiettivo. Questo cosa ci permette di dire? Ci permette di dire che io posso, innanzitutto comporre un processo, in termini di altri processi. Qui introduciamo il concetto di nesting, la possibilità di annidare elementi con altri elementi. Piccolo inciso: in Archimate, e lo vedremo negli esempi, è possibile annidare più o meno tutto; posso anche annidare, per esempio, il call center all'interno del servizio, nel momento in cui trascino call center all'interno del servizio mi troverò un servizio con all'interno il blocco call center. La relazione, però, che sussiste fra questi due elementi non viene più visualizzata. Quindi è nascosta anche se rimane. E solo un occhio esperto che sa che un'interfaccia legata a un servizio solo attraverso un operatore di una relazione di assegnamento, sa che quei due elementi sono legati attraverso l'operatore di assegnamento. Il mio consiglio, quindi, è quello di utilizzare l'annidamento solo e soltanto nel caso in cui la relazione che sussiste fra l'elemento contenitore e l'elemento contenuto, è una relazione di tipo composed off. Quindi possiamo dire, ve lo dico perché è così in questo caso, che questo processo è composto da questi altri processi. Quindi, anche se non visibili, questi processi sono legati da una relazione composed off con il processo generale rappresentato qui. Oltre a questo, a questa indicazione, che approfondiremo poi negli esercizi, cosa sto raccontando qui? Io sto raccontando che questo processo è definito da: un altro processo, potremmo chiamarlo attività, che è la ricezione dei dettagli della richiesta rispetto a quell'esame che il paziente deve fare, bisogna cercare le disponibilità e poi può accadere che o confermo la prenotazione o viene cancellata la richiesta di prenotazione. Come vedete, ho utilizzato anche in questo caso il junction per dire che cosa? O uno o l'altro, perché il junction, nella sua versione vuota, pallino vuoto, indica uno xor, o uno o l'altro. Notate anche la relazione che è stata definita fra i due elementi comportamentali, che è il trigger. Indica il fatto che l'esecuzione di questa attività triggerà quindi scatta fa scattare la possibilità di eseguire quest'altra attività. Un ulteriore elemento importante è la relazione che esiste fra il call center operator e il processo, il call center operator è responsabile del processo e di tutto quello che contiene, quindi indirettamente il call center operator è anche responsabile di questa attività, di questa attività, di questa attività, di questa attività. Ultimo aspetto, bisogna ricordarsi che non è compito di Archimate dettagliare in maniera precisa la natura di questo processo. L'obiettivo di Archimate è semplicemente dire esiste questo processo, esistono queste macrofasi, non bisogna mai cadere nella tentazione di voler definire nel dettaglio il processo. Qualcuno potrebbe dire <<eh, ma se fosse un'or invece che uno xor?>> Lasciamo perdere che qui non ha senso. Mettiamo lo xor e lo indichiamo in qualche modo. Perché? Perché non è l'obiettivo qui dire in dettaglio quello che succede per fare un'analisi anche strutturale del mio processo, comportamentale del mio processo, mi serve solo far intuire come funziona tutto. È ovvio che se metto conferma/cancel questo non può essere un or ma, deve essere un xor, perché o confermo o cancello. Se invece avessi due attività che in qualche modo si sovrappongono, è ovvio pensare che questo lo debba interpretare come un or. Quindi la tentazione non deve essere quella di descrivere in dettaglio il nostro processo. Ma il contrario è quello di dare una visione generale, perché un diagramma di Enterprise Architecture è sempre una visione generale di qualche aspetto legato alla nostra architettura. Se vogliamo il dettaglio, in questo caso specifico abbiamo BPMNE, e utilizzeremo BPMN che deve essere in qualche modo compatibile con questo processo. Ma non dobbiamo pensare di andare a generare BPMN, diciamo così, attraverso delle trasformazioni, partendo da questo e trasformandolo in un altro linguaggio, appunto in un'altra notazione, quale BPMN, perché il potere espressivo, e anche la semantica associata a questo tipo di notazione è molto più povera, potremmo dire, almeno per l'obiettivo di descrizione del processo che ha BPMN, rispetto a quella che ha BPMN. L'ultima slide riguarda l'utilizzo delle funzioni, in questo caso noi vediamo che un servizio di Examination Booking generale, che è la composizione per quello



che abbiamo detto prima, possiamo dare per scontato che sussiste una relazione di composizione fra New Examination Booking che è Cancel Examination Booking; appunto, questa examination booking è la composizione di questi due servizi, quindi questo examination booking possiamo considerarlo come un macro contenitore a disposizione del paziente, che può utilizzare sia una che l'altra; quindi il fatto che questo servizio è servito al paziente indica che anche questo e questo, sono a disposizione del mio paziente, quindi c'è una certa indirezione nella possibilità di navigare le mie relazioni. Ecco, in questo caso io cosa ho? Che dietro a questo servizio, che è quello che abbiamo appena visto, abbiamo questo processo. Infatti questo processo realizza questo servizio. Immaginiamo di aver descritto anche il servizio di cancellazione di una prenotazione già esistente. Avrà il suo processo corrispondente che lo realizza. Ecco, in questo caso, non ha senso raggruppare questi due processi con un altro processo, perché non è che ho uno e poi l'altro; sì, da un certo punto di vista è ovvio che se io faccio New, ne faccio una nuova, poi la posso cancellare, ma la posso cancellare. Non è detto che la cancelli, non è necessario avere un trigger fra uno e l'altro, potrebbe esserci, potrebbe non esserci, non è nemmeno utile andare a mettere una junction per dire sì, potrebbe esserci, così come potrebbe non esserci, ma il potrebbe non esserci. che attività poi metto qua? Non esiste il termine attività. Per questo motivo è molto più utile dire, guardate che questi due elementi comportamentali che corrispondono ai due processi che realizzano questi due servizi, fanno tutti capo a una funzione della mia azienda che è l'Examination Booking, Call Center, di cui è responsabile il call center operator; quindi il call center operator, è responsabile dell'intera funzione e quindi di tutti gli elementi comportamentali che stanno al loro interno. Nelle prossime, nella prossima lezione vedremo di approfondire tutti questi concetti con degli esempi pratici, cercando di, partendo dal pattern di base, andare ad arricchirlo per creare qualcosa di significativo e qualcosa di interpretabile, per rappresentare appunto dei servizi offerti da organizzazioni. Poi, per la parte più implementativa, come un sistema informativo sostiene questo tipo di servizio, andremo, quando sarà il momento, ad approfondire anche il livello applicativo, poi quello tecnologico.

