

# **Metodologia per la Formalizzazione dei Requisiti di Produzione**

# **Metodologia per la Formalizzazione delle Caratteristiche dei Sistemi Produttivi**

*OR 1: Configurazione e Gestione di Sistemi  
Integrati di Produzione*

*Responsabile Area 1: prof. ing. Tullio Tolio  
Ricercatore: ing. Andrea Polato*

## Indice

1. Introduzione .....	3
2. Stato dell'arte .....	4
2.1. Literature Review .....	4
3. Obiettivi della Ricerca .....	6
4. Criteri di Modellazione .....	7
5. Il Reference Framework Concettuale .....	7
5.1. Production system sub-package (R1.2).....	8
5.2. Product sub-package (R1.1) .....	11
5.3. Process sub-package (R1.1) .....	12
5.4. Management sub-package (R1.2) .....	14
5.5. Integrazione: La classe Transformation.....	15
6. Prime applicazioni .....	15
6.1. Linea flessibile che produce piastre portanti per alberi a camme.....	15
6.2. Il Problema di Configurazione del Sistema Produttivo.....	18
7. Attività svolte.....	23
7.1. Grado di avanzamento.....	23
8. Conclusioni e Sviluppi futuri .....	23
9. Bibliografia .....	23

## Premessa

Il presente documento raggruppa entrambi i report previsti per il dodicesimo mese di attività relativamente all'OR 1 ("Configurazione di Sistemi Integrati di Produzione") e cioè il report R1.1- "Metodologia per la formalizzazione dei requisiti di produzione" e il report R1.2 - "Metodologia per la formalizzazione delle caratteristiche dei sistemi produttivi". La scelta di presentare un documento unico è stata dettata dalla forte relazione che lega i due argomenti e dalla convinzione che la trattazione relativa la formalizzazione delle informazioni di prodotto, processo e sistema non possa avvenire considerando separatamente i tre domini. Ogni sezione del documento è pertanto da considerarsi comune ai due argomenti, le parti che riguardano specificatamente il primo report sono le sezioni 5.2 e 5.3, mentre quelle che relative al secondo sono le sezioni 5.1 e 5.4.

## 1. Introduzione

Il settore manifatturiero è caratterizzato da un sempre maggior grado di complessità, dovuto sia all'elevato numero di requisiti tecnici che devono essere soddisfatti a livello produttivo, sia dalla presenza di diverse fonti di incertezza nel mercato.

Questo elevato grado di incertezza condiziona sia gli aspetti fisici che architetturelle delle industrie manifatturiere oltre a quelli manageriali, finanziari e organizzativi. La crescente complessità dei requirements e dell'ambiente competitivo rende estremamente critici i problemi di configurazione, riconfigurazione, implementazione, gestione, controllo e miglioramento continuo dei prodotti, processi e sistemi produttivi. Per gestire una tale complessità è necessario considerare le principali relazioni tra i diversi oggetti che compongono l'intero sistema, adottandone una visione olistica e fortemente integrata. A tale scopo, gli ingegneri di sistema hanno sviluppato, nel corso del tempo, diversi strumenti e metodologie di analisi basate sulla conoscenza per supportare i processi decisionali durante tutto il ciclo di vita del prodotto, processo e sistema. Tuttavia, a seconda della fase del ciclo di vita in cui ogni decisore si trova ad operare, sono necessari diversi livelli di dettaglio delle informazioni richieste e le metodologie e gli strumenti sviluppati a supporto di tali processi decisionali devono tener conto di queste diverse esigenze.

Nella fase di configurazione di un sistema produttivo a partire "da prato verde", sono disponibili per esempio informazioni molto aggregate sui requisiti di produzione e sulle capabilities. In questa fase, tecniche analitiche esatte [1] [2] sono usate molto frequentemente per supportare il processo decisionale; queste tecniche richiedono infatti poche ed aggregate informazioni sul prodotto, il sistema e il processo produttivo e forniscono delle configurazioni generali che sono sufficientemente adeguate per far fronte alle richieste produttive. D'altra parte, le tecniche analitiche approssimate [3] [4] sono più appropriate durante la fase più dettagliata di riconfigurazione, quando è già disponibile una certa conoscenza sulla dinamica del sistema e sugli eventi inattesi che ne influenzano le prestazioni. Questo genere di tecniche ha bisogno tuttavia di informazioni più precise riguardo sia il comportamento dinamico del sistema, sia i nuovi requisiti di produzione, e fornisce in output, dopo aver valutato l'insieme delle misure di prestazione del sistema, un set ammissibile di configurazioni di dettaglio che soddisfano i requisiti di partenza. Quando si ha a che fare con problemi di gestione del sistema produttivo, il livello delle informazioni disponibili è ancora più completo e dettagliato come nel caso di problemi di dispatching dei prodotti o di loading delle macchine. Generalmente, per questi problemi, il decisore è supportato dalla simulazione [5] [6] che permette di quantificare le variazioni delle prestazioni del sistema al variare delle politiche di gestione. Infine, quando si controllano e si migliorano le prestazioni di un sistema produttivo, il livello di dettaglio è ancora maggiore, perché sono necessarie informazioni specifiche sul comportamento di ogni singola risorsa e di ogni singolo processo. Per questi problemi, sono state introdotte e vengono comunemente utilizzate tecniche e metodi statistici quali lo SPC (Statistical Process Control) [7] o il DOE (Design of Experiments) [8].

Per essere effettivamente utilizzate in modo integrato ed armonico sia dalle imprese che dai ricercatori, tutte le metodologie e gli strumenti precedentemente accennati dovrebbero condividere lo stesso modello concettuale. Infatti, uno dei problemi più difficili e mai risolti in produzione è la mancanza di un framework per la gestione delle informazioni produttive, che sia in grado:

- di supportare l'utente nell'attività di modellazione del sistema produttivo, durante tutte le diverse fasi del ciclo di vita;
- di fornire agli strumenti utilizzati dal decisore come supporto decisionale, esattamente le informazioni che questi necessitano.

I principali requisiti che dovrebbe soddisfare un tale framework per la gestione della conoscenza sono descritti con i termini *flessibilità*, *estendibilità*, *scalabilità* [9] e *integrazione*.

- *Flessibilità*: il modello deve essere facilmente adattabile in modo da poter descrivere diverse architetture del sistema produttivo, processi e caratteristiche di prodotto.
- *Scalabilità*: il modello deve potere supportare la descrizione del prodotto, del sistema e del processo produttivo a diversi livelli di dettaglio.
- *Estendibilità*: il modello deve garantire all'utilizzatore la possibilità, se necessario, di estendere velocemente il livello di dettaglio.
- *Integrazione*: i prodotti, i processi ed i sistemi produttivi, oltre alle relative relazioni, devono essere considerati e descritti nello stesso framework, poiché essi sono tutti parte dell'ambiente produttivo.

## 2. Stato dell'arte

### 2.1. Literature Review

In letteratura, il problema di sviluppare un framework concettuale per la gestione delle informazioni produttive è stato tradizionalmente affrontato proponendo diverse soluzioni che possono in alcuni casi essere adattate facilmente alle singole situazioni, ma che non saranno mai "complete", in quanto non considerando tutti i requisiti definiti nel paragrafo precedente. Lo svantaggio principale è che i modelli attuali considerano generalmente i prodotti, i sistemi e i processi produttivi come separati l'uno dall'altro.

Con riferimento al sistema produttivo, molti lavori riguardano l'utilizzo di linguaggi object-oriented per lo sviluppo di modelli concettuali. In questi lavori, il modello del sistema produttivo è basato sulla decomposizione del sistema in oggetti a loro volta raggruppati in classi. Ogni oggetto ha un'identità, uno stato e un comportamento coerentemente con il paradigma object-oriented. Il sistema reale è considerato da tre punti di vista: quello fisico, quello funzionale e quello dinamico [10]. Lo svantaggio principale delle metodologie orientate agli oggetti è stato, fino alla creazione, intorno alla fine degli anni 90, di un linguaggio standardizzato, la confusione generata dall'utilizzo di diversi linguaggi da parte dei singoli "modellatori" [11]. Alcuni esempi sono i seguenti.

Park et al. [12] propongono un framework di modellazione orientato agli oggetti denominato JR-net per un generico AMS (Automated Manufacturing System). In questo caso i tipi di oggetto "resource", "job" e "control" ne compongono il modello. Si considerano tre livelli di modellazione che sono denominati rispettivamente "layout model" level, "functional JR-net model" level e "control model" level. Il primo livello si riferisce all'aspetto architeturale del sistema; nel secondo i vari job sono collegati alle risorse necessarie per la loro produzione e infine nel terzo livello è modellata l'assegnazione condizionata dei job alle risorse.

Kellert et al. [13] propongono un modello concettuale per FMSs (Flexible Manufacturing Systems), basato ancora su una decomposizione a tre livelli: logico, fisico e decisionale. M2PO è usato come linguaggio di modellazione, mentre il metodo di SADT (Structured Analysis and Design Technique) è utilizzato per specificare i flussi delle informazioni e del materiale.

Liu et al. [14] usano linguaggi di modellazione diversi per creare i modelli degli stessi tre livelli di analisi del sistema. Per esempio, per rappresentare la struttura statica del sistema vengono utilizzati differenti

diagrammi che sono l'association, l'aggregation e il propagation diagram. Come prima, DFD (Data Flow Diagram) è usato per modellare la struttura dinamica del sistema. In questo lavoro, il livello gestionale non è invece trattato in dettaglio.

Booch et al. [15] propongono un altro modello object-oriented per FMSs. Gli autori usano il formalismo OMT (Object Modelling Technique) per modellare la vista statica del sistema, il DFD per i modelli dinamici e funzionali e lo STD (State Transition Diagram) per gli aspetti di controllo.

Questi papers rappresentano alcuni dei lavori più importanti dell'era pre-UML, cioè sono stati pubblicati prima che lo standard UML (Unified Modelling Language) per la modellazione ad oggetti fosse definito ed accettato come linguaggio di modellazione unitario.

Tra i metodi dell'era di UML, Bruccoleri et al. [11] e Matta et al. [10] propongono un approccio di modellazione basato su UML per descrivere tutte le funzioni statiche e dinamiche rispettivamente di un controllore di cella e di un FMS completo. In particolare, il lavoro di Matta et al. [10] è stato sviluppato nel contesto di un progetto nazionale italiano del FIRB dal nome "Software frameworks and technologies for the development and maintenance of open-source distributed simulation code, oriented to the manufacturing field" in cui il Dipartimento di Tecnologie e Sistemi di Lavorazione del Politecnico di Milano era particolarmente coinvolto a livello di modellazione.

Altri contributi interessanti per la letteratura dell'era UML sono i documenti scritti da Van Brussel ed al. [16] [17], nei quali i class diagrams di UML sono usati per presentare un'Holonic Reference Architecture per i sistemi produttivi, sviluppata negli anni 90 al KU di Leuven, Belgio. Questa architettura è basata su un paradigma di controllo di alto livello che facilita il costante adattamento dei sistemi produttivi ai frequenti cambiamenti nei requisiti di produzione.

Ancora sulla modellazione orientata agli oggetti dei sistemi produttivi, [18] presenta una review dettagliata sugli esistenti formalismi object oriented di larga scala e sui relativi strumenti specificamente sviluppati per la simulazione dei sistemi produttivi. In particolare, il lavoro analizza i seguenti metodi: BLOCS/M, DEVS, Laval, OOSIM, OSU-CIM e SmatSim/SmarterSim.

Oltre gli sforzi di ricerca che vengono dall'accademia, anche le imprese hanno proposto negli anni scorsi molti contributi interessanti nell'ambito della modellazione dei sistemi produttivi. Fra tutti, un metodo molto vicino al fondamento logico del presente lavoro è stato presentato per la prima volta dall'italiana MCM (Machining Centers Manufacturing), uno dei principali costruttori mondiali di macchine utensili, nel corso del meeting del Gennaio 2006 del CIRP, in occasione della prima riunione del CIRP "SPECIES-Production System Evolution" Working Group. Il metodo sviluppato è basato su un framework orientato agli oggetti per la gestione della conoscenza nel dominio dei sistemi di produzione riconfigurabili e flessibili. Questo reference model ha l'obiettivo pratico di supportare il processo di progettazione del sistema, sia da un punto meccanico che di controllo, con un approccio fortemente integrato [19].

Per quanto riguarda la formalizzazione delle informazioni di prodotto, un contributo interessante di Krause ed al. [20] presenta un'overview dello stato dell'arte e delle pratiche di product modeling, in termini di modelli del prodotto e catene di processi.

Molti importanti modelli delle informazioni di prodotto sono forniti dagli standard industriali che sono stati sviluppati nel corso degli anni nei diversi ambiti di ricerca quali ad esempio il PLM (Product Lifecycle Management). Alcuni degli standard più importanti sono STEP (ISO 10303) e PLCS (ISO 10303-239). Nuovi sviluppi in quest'area sono tutt'ora in fase di sviluppo da parte del FP6 Integrated Project chiamato PROMISE (PROduct lifecycle Management and Information tracking using Smart Embedded systems), in cui si sono mossi i primi passi verso un nuovo standard aperto che copre l'intero set di fasi del ciclo di vita del prodotto. Per una prima reference su questo approccio si faccia riferimento a [21].

Muovendosi verso il campo della formalizzazione della conoscenza di processo, un approccio interessante è quello del progetto PSL (Process Specification Language) (ISO/CD18629 2002) [22], che mira a sviluppare un'ontologia generale per rappresentare i processi produttivi e per lo scambio di informazioni di

processo e di conoscenza. PSL fornisce un linguaggio per lo scambio di dati di processo con lo scopo di integrare applicazioni multiple relative ai processi durante il ciclo di vita del contesto produttivo.

Focalizzandosi poi sugli aspetti di integrazione fra prodotto e processo, lo standard STEP-NC (ISO 14649) [23] presenta un modello di interoperabilità di dati fra sistemi CAD/CAM e le macchine CNC.

Infine, per quanto riguarda l'integrazione delle informazioni sui prodotti, i processi e i sistemi produttivi, il prof. Kimura [24] propone un framework di modellazione di prodotto e processo da un punto di vista produttivo virtuale.

Martin et al. [25] propongono uno strumento chiamato "Ontoforge", per supportare la progettazione integrata di un prodotto forgiato considerando la conoscenza di processo e le informazioni relative alle risorse che realizzano il processo. Il dominio di formalizzazione della conoscenza è tuttavia limitato alla forgiatura.

Ortega et al. [26] presentano un meta-modello usando il formalismo Express G per includere STEP in un dominio di produzione flessibile.

Lavori più orientati all'integrazione a livello d'impresa sono quello di Harding et al. [27], che descrive un'ontologia per supportare le squadre di produzione situate in diverse aree del globo nel condividere una comprensione comune degli aspetti produttivi, per raggiungere l'interoperabilità e per riutilizzare la conoscenza acquisita. Gli autori estendono a livello di rete globale il loro lavoro precedente [28], che riguardava lo sviluppo di un moderatore MSE (Manufacturing System Engineering) da utilizzarsi in una singola impresa.

Altri contributi importanti nel campo della formalizzazione della conoscenza per l'enterprise integration si trovano in [29], [30], [31] e [32] dov'è presentato il linguaggio CIMOSA per la modellazione delle informazioni di sistema.

Per concludere questa breve descrizione dei contributi presenti in letteratura, si sottolinea che nessuno dei lavori esistenti sembra poter rappresentare insieme i dati di prodotto, processo e sistema, le informazioni e la conoscenza, rispondendo alle esigenze evidenziate per il progetto in esame.

### **3. Obiettivi della Ricerca**

In conformità con quanto definito nel Piano Attuativo relativamente all'obiettivo 1, lo scopo del lavoro di ricerca effettuato è stato quello di proporre un linguaggio di modellazione object oriented in grado di formalizzare la conoscenza di prodotto, processo e sistema produttivo. Tale formalismo rispetta i requisiti di flessibilità, scalabilità, estendibilità e integrazione definiti come obiettivi del progetto. Il linguaggio di modellazione adottato è UML (Unified Modeling Language) ed in particolare il noto formalismo del class diagram. Il modello è stato sviluppato per soddisfare le esigenze di utenti diversi, quali i gruppi di ricerca con le relative competenze multidisciplinari, i responsabili della produzione, i progettisti di sistema e i vendors di sistemi produttivi. Lo standard STEP NC, focus della ricerca nei primi sei mesi di attività (si veda Rendiconto Semestrale) ha rappresentato il primo riferimento fondamentale per la modellazione della conoscenza di prodotto e di processo. In particolare lo step successivo è stato quello di tradurre il relativo framework concettuale nel formalismo del class diagram di UML e di integrarlo nel diagramma generale con cui si sono modellate le informazioni di sistema. Questo ha consentito quindi di modellare nello stesso framework le tre aree del dominio produttivo considerando congiuntamente le interazioni tra loro.

Nella sezione seguente sono brevemente illustrati i criteri di modellazione utilizzati, motivando la scelta del linguaggio di modellazione e focalizzandosi sulle caratteristiche principali del modello stesso. Nel paragrafo 5, è presentato in dettaglio il modello concettuale; a partire da una visione generale, ogni singolo elemento del modello viene descritto con precisione. Nel paragrafo 6, sono presentate le prime applicazioni del framework proposto, con lo scopo di mostrare i vantaggi di tale metodo e della relativa applicabilità ai casi reali. Infine, nel paragrafo 7 si riportano alcune conclusioni e gli sviluppi futuri che verranno realizzati a partire dalle linee guida delineate dal referente framework presentato.

## 4. Criteri di Modellazione

La prima necessità dell'attività di modellazione era di definire più precisamente le finalità di modellazione dei diversi utenti e soprattutto, i problemi che essi devono affrontare e risolvere grazie all'utilizzo di questi modelli durante il lavoro di tutti i giorni. Questa attività ha portato ad una comprensione di base dei requisiti di modellazione presentati dai diversi utenti ed ad una documentazione testuale non formalizzata di un insieme specifico di casi d'uso. A partire da questi casi d'uso, il passo seguente è stato quello di muoversi verso un'astrazione dei requisiti raccolti, per sviluppare un meta-modello in grado di convertirsi facilmente in un framework concettuale di riferimento che fosse adottabile trasversalmente da tutti gli utenti identificati.

Particolare importanza è stata data alla vista statica degli oggetti da modellare, cioè alla descrizione dei loro componenti architettonici, dei loro attributi e delle relazioni tra essi. In particolare si è deciso di adottare UML (Unified Modeling Language), lo standard *de facto* per la modellazione di sistemi (principalmente software), come linguaggio di modellazione, per l'ampia possibilità di modellazione di tutti i diversi aspetti dei sistemi complessi quali sono i moderni sistemi produttivi automatizzati. Per ulteriori dettagli riguardo UML si faccia riferimento a [33] [34].

Il nucleo del framework di riferimento è un class diagram che rappresenta i prodotti, i processi ed i sistemi produttivi. Il formalismo del diagramma delle classi consente al modellatore di descrivere le caratteristiche principali di un sistema, in termini di oggetti che lo compongono, di attributi che descrivono le caratteristiche principali di questi oggetti e di inter-relazioni che esistono fra loro (si veda la sezione seguente per una descrizione dettagliata).

La natura orientata agli oggetti del framework è una caratteristica fondamentale del modello, in ottica di una facile riutilizzazione degli elementi del modello da parte degli utenti target, e in riferimento al raggiungimento degli obiettivi del progetto (paragrafo 1 e 3). L'orientamento agli oggetti del modello è spiegato dalle seguenti caratteristiche del framework di modellazione, caratteristiche che sono comuni a tutti i modelli realmente object oriented:

- *astrazione della classificazione*: permette la descrizione degli oggetti, solo nei termini che sono realmente interessanti per il problema, trascurando i particolari secondari;
- *incapsulazione*: la possibilità di nascondere i particolari inutili relativi all'implementazione di un dato oggetto;
- *modularità*: ciò significa che gli elementi da cui il modello è sviluppato sono fortemente disaccoppiati ma allo stesso tempo consistenti uno con l'altro;
- *ereditarietà*: si riferisce alla possibilità di specificare le gerarchie tra gli elementi del modello;
- *aggregazione*: si riferisce alla possibilità di definire un oggetto come insieme di altri oggetti.

## 5. Il Reference Framework Concettuale

In questa sezione viene presentato in dettaglio il modello UML per un ambiente generico di produzione in cui prodotto, sistema e processo produttivo interagiscono. Seguendo la struttura usuale di un diagramma UML, il modello è stato sviluppato per essere suddiviso nelle tre aree di massimo interesse: la model management area, la structural area e la dynamic area. Ogni area rappresenta un punto di vista specifico dell'ambiente modellato ed è caratterizzata dal relativo insieme specifico di diagrammi. In questa sezione, ci si concentra sulla Structural Area in cui vengono discussi gli oggetti che appartengono al sistema. Da questo punto di vista, sia elementi fisici (buffers, prodotti, macchine, tools, ecc.) che concetti astratti (trasformazione, strategia, piano di produzione) sono modellati con gli strumenti più comuni per la modellazione ad oggetti quali associazioni, composizioni, generalizzazioni, ecc. Il diagramma più importante in quest'area è il class diagram, che sarà presentato alla fine di questa sezione.

Il reference framework concettuale è rappresentato in figura 1. In particolare, viene riportato il diagramma delle classi di UML che rappresenta le interazioni tra prodotto, processo e sistema produttivo da un punto di vista statico. Come si può notare da questo schema, è possibile distinguere quattro sottogruppi principali di classi, chiamati sub-packages, ognuno dei quali si concentra su aspetti diversi del dominio produttivo. Queste quattro aree verranno chiamate in seguito *Production System sub-package*, *Product sub-package*, *Process sub-package* e *Management sub-package*. Questi sub-packages non sono considerati separatamente, ma vengono invece considerate direttamente le interazioni fra di essi e rappresentate nel framework seguendo le regole del formalismo di UML. Il nucleo dell'integrazione fra prodotto, processo e sistema produttivo è la *TRANSFORMATION class* che assume un'importanza rilevante in quanto rappresenta il collegamento fra i quattro sub-packages. Nei seguenti 5 sottoparagrafi, ogni sub-package sarà presentato in dettaglio.

### **5.1. Production system sub-package (R1.2)**

Il nucleo del production system sub-package è il *SYSTEM class*, che definisce l'insieme dei sottosistemi fisici e/o delle risorse da cui il sistema è costituito e raggruppato, secondo alcuni legami logici, quali le proprietà architetture, la natura delle trasformazioni tecnologiche effettuate, l'ubicazione fisica nel sistema o altri aspetti di modellazione. Il sistema produttivo è rappresentato nel modello da un oggetto del *SYSTEM class*, che a sua volta si compone di almeno una risorsa (cioè di almeno un oggetto del *RESOURCE class*) e/o di uno o più sottosistemi (cioè oggetti del *SYSTEM class*). Inoltre, esso deve essere collegato almeno ad una trasformazione, cioè almeno ad un oggetto del *TRANSFORMATION class*.

La possibilità di creare gerarchicamente dei sottosistemi generando “un anello chiuso” sul *SYSTEM class* tramite l'associazione di aggregazione rappresentata nel diagramma, è introdotta per soddisfare, almeno a livello di sistema, i requirements di scalabilità, flessibilità e estendibilità del modello sviluppato. Infatti, questo schema logico consente di generare modelli di sistemi produttivi con diversi livelli di dettaglio, secondo la lista specifica dei requisiti definiti dall'utente. Inoltre, ad ogni livello, tutti i “pezzi” di informazione contenuti nel *SYSTEM class* possono essere stanziati come necessari, così garantendo la flessibilità necessaria del modello, non forzando l'utilizzatore ad aggiungere inutili particolari secondari. Un primo esempio di utilizzo di questo modello è fornito nel paragrafo 6.

Il *SYSTEM class* può essere quindi utilizzato a diversi livelli di dettaglio. A seconda dell'interesse del modellatore, possono quindi essere stanziati oggetti del *SYSTEM* e/o del *RESOURCE class* che si posizionano a diversi livelli differenti della gerarchia architetture. E' possibile quindi formalizzare attraverso le classi e le associazioni definite, tutte le informazioni che riguardano per esempio la posizione del sottosistema all'interno dell'intero layout e l'integrazione con altri sottosistemi oltre alle informazioni riguardanti gli aspetti tecnologici e le proprietà fisiche del sistema/sottosistema.





Il *RESOURCE* class modella ogni tipo di risorsa che compone il sistema. Ci sono cinque tipi di risorse nel modello: macchine, buffers, pallets, attrezzature e tools. Si possono modellare le informazioni riguardanti la posizione di ogni risorsa nel sistema, le connessioni fisiche e logiche fra le risorse e tutti gli altri aspetti architettonici. Inoltre, anche i parametri di affidabilità, prestazione e costo fanno parte della lista di attributi delle classi.

Entrando nei particolari dei diversi tipi di risorse, gli oggetti della classe *MACCHINE* sono definiti come le risorse che realizzano trasformazioni di specie e di spazio sui prodotti, sugli attrezzi e sui pallet (questi ultimi due sono a loro volta risorse! - ulteriori particolari sono forniti in seguito). Nel presente modello, si possono modellare non solo le informazioni generali sulle macchine, ma si possono aggiungere informazioni specifiche e più dettagliate su uno o più tipi di macchina estendendo il modello con relative sotto-classi della classe *MACHINE* e opportuni attributi. Per esempio, può essere aggiunto un nuovo *NC\_Machining\_Centre*, con attributi quali la definizione del relativo cubo di lavoro, il numero degli assi controllati, l'avanzamento, la velocità massima per ogni asse, ecc.

Gli oggetti della classe *BUFFER* definiscono quelle risorse che possono realizzare solamente trasformazioni di spazio (movimentazioni) sul prodotto, gli utensili e i pallets. Esempi di tali buffers sono i conveyors, gli AGVs, i carrelli, altri elementi di trasporto, ecc. Anche per la classe *BUFFER*, è possibile aggiungere nuove sottoclassi che descrivono, con tutti i dettagli necessari, i diversi tipi di buffer che esistono nei sistemi produttivi comuni. Questi tipi di buffers possono inoltre essere classificati per esempio in termini di proprietà funzionali (buffer omogenei e non omogenei, dedicati o condivisi, statici o dinamici) o in termini di caratteristiche fisiche (buffer per gli utensili, per i prodotti, per le informazioni, per i pallet, ecc.).

La resource *PALLET* è una risorsa particolare, a seconda della volontà del modellatore. Infatti, in un sistema produttivo, il pallet è l'interfaccia fisica standard tra le risorse di sistema e le parti da lavorare. Il pallet ha la funzione di trasportare le parti che devono essere processate dalle macchine che compongono il sistema. I prodotti sono fissati sui pallet attraverso attrezzature standardizzate che hanno la funzione di assicurare la stabilità alle parti durante la lavorazione.

È interessante sottolineare che, nel modello attuale, i *pallet* ed *attrezzature* sono considerati a tutti gli effetti delle *resources*, in quanto sono dispositivi fisici che supportano il sistema nel realizzare le trasformazioni sui prodotti, effettuando determinati processi di produzione. Tuttavia sono particolari risorse, poiché anch'essi subiscono delle trasformazioni da parte dei buffer e delle macchine del sistema.

Le classi *PALLET* e *FIXTURE* (attrezzatura) contengono tutte le informazioni tecnologiche che descrivono questi oggetti. Inoltre, la classe *PALLET* contiene le informazioni riguardanti la codifica dei tipi di parte caricate su di esso, in un dato istante temporale. Gli oggetti di questa classe sono direttamente collegati gli uni agli altri tramite l'utilizzo dell'associazione di aggregazione. Ciò significa che un pallet è visto come aggregazione di una o più attrezzature. Sia le classi *PALLET* che *FIXTURE* sono associate ad un particolare *SETUP*. Questa classe rappresenta il collegamento tecnologico fra product, process e system subpackages e verrà descritta nelle seguenti sottosezioni.

Questo modo di modellare l'interazione fra i pallets e le attrezzature consente di descrivere adeguatamente sia FMSs (Flexible Manufacturing Systems) che DFLs (Dedicated Flow Lines). Infatti, generalmente, nelle DFLs, che realizzano un insieme di operazioni in stretta sequenza e con elevati volumi di produzione, una sola parte è posizionata su un pallet e questo è a sua volta personalizzato per la geometria di quel specifico tipo di parte. Inoltre, in questi sistemi è generalmente presente solo un tipo di pallet. Lo stesso vale solitamente anche per le attrezzature necessarie. Ne consegue, che le classi *PALLET* and *FIXTURE* abbiano un'importanza relativa nel caso di DFLs. Per quanto riguarda gli FMSs, che processano normalmente una grande varietà parti, diversi tipi di parte possono invece essere caricati sullo stesso pallet, tramite l'utilizzo di attrezzature diverse. Di conseguenza le classi *PALLET* e *FIXTURE* assumono un'importanza particolare in questo caso e dovrebbero perciò essere utilizzati nella maggior parte delle applicazioni del modello.

L'ultimo tipo di risorsa del framework di riferimento è modellata dalla classe *TOOL*. Gli utensili sono quelle risorse che fungono da collegamento fisico fra il sistema ed il prodotto, durante la realizzazione del

processo produttivo. Gli attrezzi contengono le informazioni che sono rigorosamente necessarie per il processo di lavorazione che deve essere realizzato. La classe *TOOL* definisce in particolare le informazioni riguardo l'insieme degli utensili utilizzati dal sistema, i magazzini in cui sono riposti, la vita residua dello stesso e il tipo di lavorazione che possono realizzare. *TOOL* infatti è una classe di confine che collega il sub-package del sistema produttivo al sub-package del processo (si vedano le sottosezioni seguenti).

Il System sub-package appena descritto è stato utilizzato per descrivere molti tipi di sistema produttivo da un punto di vista strutturale. In particolare, il framework ha mostrato buone capacità di modellazione, sia nel caso di contesti di produzione rigida, quali le linee transfer, sia per contesti flessibili/riconfigurabili, quali FMSs e RMSs (Reconfigurable Manufacturing Systems).

## 5.2. Product sub-package (R1.1)

Il product sub-package rappresenta la parte superiore destra del Reference Framework proposto (figura 1). Poiché in letteratura sono già disponibili diversi standard per la formalizzazione della conoscenza di prodotto (si veda la sezione relativa), l'idea non è stata quella di sviluppare un nuovo modello, piuttosto quella di selezionare, tra gli standard esistenti, quello che meglio si adatta agli obiettivi/requisiti del framework attuale. In questo senso, poiché il modello sarà principalmente usato durante la fase di design di un contesto produttivo, viene da se l'utilizzo dello standard industriale STEP-NC (ISO 14649).

Inoltre, poiché il modello proposto include anche una parte chiamata "Process sub-package", STEP-NC rappresenta un ottimo riferimento da adottare dato che include ampiamente anche aspetti di processo. Di conseguenza, gli sforzi verso lo sviluppo del product sub-package consistono principalmente nel tradurre il framework concettuale di STEP-NC, formalizzato nel linguaggio EXPRESS G, nel formalismo del class diagram di UML e successivamente nel collegare le classi del sub-package con gli altri sub-packages del modello, in una vista assolutamente coerente.

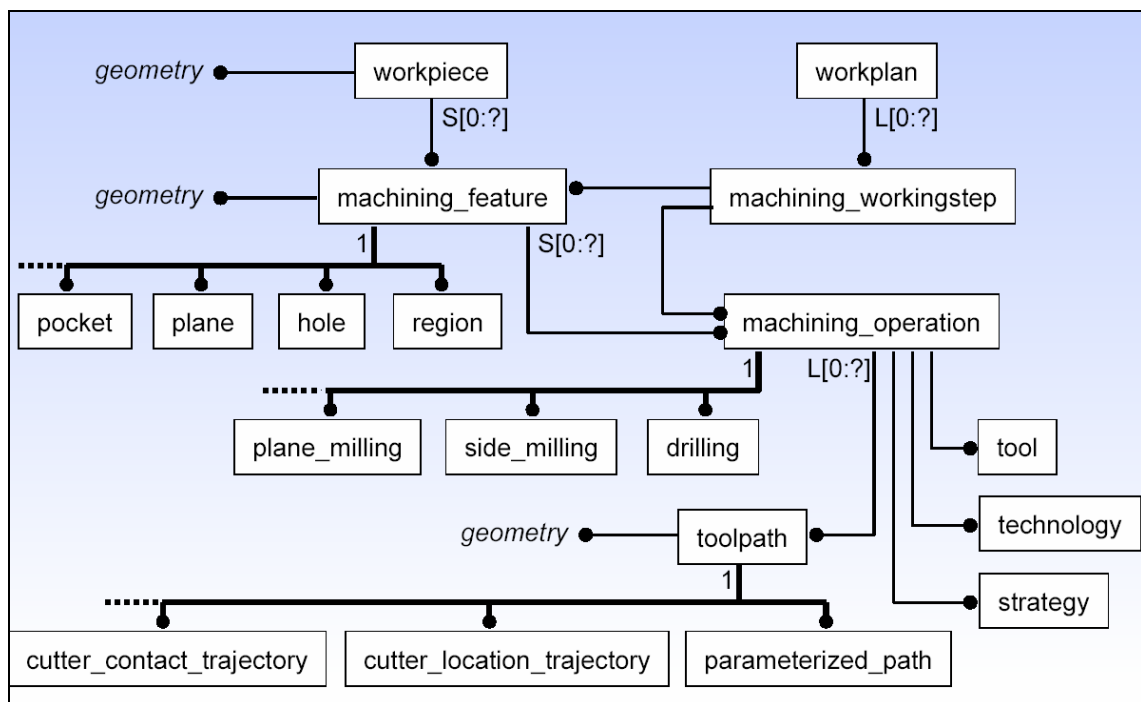


Fig. 2 Class diagram del Reference Framework Concettuale

Secondo STEP NC, la classe principale del product sub-package è la classe *WORKPIECE*. Un oggetto di questa classe è definito come il tipo di prodotto realizzato dal sistema ed è direttamente in rapporto di uno a uno con il codice del prodotto come nel caso degli ordini da cliente. Un pezzo in lavorazione inoltre si riferisce al prodotto nella relativa forma “as-designed”.

Ancora coerentemente con STEP NC, un oggetto in questa classe contiene tutte le informazioni riguardo il materiale del prodotto, le specifiche relative alle geometrie del grezzo e del finito, le bounding geometry, le tolleranze, così come i dettagli sui diversi tipi di costo. Ogni pezzo in lavorazione è associato ad uno o più prodotti fisici, cioè istanze di prodotto, che sono modellate dalla classe *PRODUCT*. Questa classe contiene tutte le informazioni osservate e misurate relative al prodotto fisico di un determinato tipo, quali le caratteristiche geometriche del finito, i risultati delle prove effettuate, ecc. E' importante sottolineare la differenza fra il significato concettuale di queste due classi. Per fornire un'idea più chiara, se la classe *WORKPIECE* rappresenta la specifica del progetto del prodotto, classe *PRODUCT* fornisce l'immagine di soltanto uno specifico prodotto di un determinato tipo. Di conseguenza, la classe *PRODUCT* è direttamente collegata alla classe *TRANSFORMATION*, poiché ciò che è realmente soggetto a cambiamenti di stato durante la lavorazione è il prodotto fisico stesso e non il prodotto come progettato (*WORKPIECE*).

Seguendo STEP NC, ogni *WORKPIECE* è descritto da una serie di features, ognuna delle quali è modellata dalla classe *MANUFACTURING FEATURE*. In STEP NC, il blocco elementare di interazione prodotto-processo è il *MACHINING WORKINGSTEP* che è formato da una *MANUFACTURING FEATURE* e da una *MACHINING OPERATION*. La manufacturing feature contiene di tutte le informazione che descrivono la superficie da lavorare, mentre la machining operation contiene tutti i pacchetti di informazione relativi al processo per una parte specifica del pezzo in lavorazione, dettagliati dalle informazioni sulla relativa feature.

Di conseguenza, la classe *MANUFACTURING FEATURE* contiene una classificazione secondo il tipo di feature, quale una superficie piana, una tasca, una scanalatura, un gradino, un foro cilindrico, un percorso utensile, un profilo, una svasatura, un filetto. Inoltre, contiene l'insieme di tutte le operazioni connesse alla feature e richieste per realizzarla. Si noti che le operazioni non sono necessariamente eseguite una dopo l'altra. Per concludere, *MANUFACTURING FEATURE* indica sia il pezzo in lavorazione di cui fa parte, sia tutte le informazioni rilevanti per realizzare la feature sul workpiece. Un *WORKPIECE* è così un'aggregazione di una o più *MANUFACTURING FEATURE*.

Concludendo la descrizione del product sub-package, i workpieces di tipi differenti possono aggregarsi per formare un lotto di produzione (classe *PRODUCTION LOT*). L'introduzione di questo concetto e della classe *PRODUCTION LOT* relativa ad esso, diventa importante quando si ha a che fare con problemi di management, quali lo scheduling e il sequenziamento. Un lotto di produzione è infatti un insieme di prodotti (tipi e volumi) che devono essere prodotti nel sistema in un dato orizzonte di tempo. La classe *PRODUCTION LOT* contiene le informazioni per quanto riguarda il tipo di prodotti che compongono il lotto, i requisiti di volume di ogni tipo di prodotto, le date di rilascio e di consegna. Inoltre, diversi lotti di produzione possono comporre un programma di produzione (classe *PRODUCTION PLAN*). Un programma di produzione consiste nella sequenza di lotti che devono essere prodotti all'interno di un turno produttivo, per soddisfare la domanda nel rispetto dei vincoli di capacità. La classe *PRODUCTION PLAN* descrive l'insieme e la sequenza di lotti di produzione da evadere, oltre alle date di inizio e di fine del piano.

Questa classe completa la descrizione del product sub-package: questo modello concettuale del prodotto è solo una delle infinite possibilità di formalizzare le informazioni di prodotto che possono essere adottate. Il framework proposto è stato sviluppato con enfasi particolare sugli aspetti del modello che ne garantiscono l'utilizzo come tool di modellazione nel contesto di design/re-design e gestione del sistema di produzione. Ulteriori sforzi saranno dedicati in futuro per includere nel modello aspetti orientati al ciclo di vita del prodotto.

### **5.3. Process sub-package (R1.1)**

Come già sottolineato nella sezione precedente, lo standard STEP NC è stato adottato per formalizzare nel reference framework proposto le informazioni di prodotto e processo. Tuttavia, STEP NC è fino ad ora in grado di modellare soltanto un numero limitato di processi produttivi quali fresatura, tornitura e processi

EDM. Queste capacità si sono dimostrate sufficienti per provare l'efficacia del framework proposto in molti casi dell'area manufacturing. Tuttavia, questo rappresenta certamente una limitazione del corrente stato dell'arte del modello STEP NC.

La classe principale del process sub-package è la classe *MACHINING WORKINGSTEP*. Un machining workingstep costituisce il building block elementare di interazione tra prodotto e processo e rappresenta il processo di lavorazione per una parte specifica della superficie del pezzo da lavorare. Un machining workingstep non può esistere senza una feature da realizzare ed un'operazione di lavorazione. Quest'ultima è modellata dalla classe *MACHINING OPERATION*. La feature raccoglie tutte le informazioni geometriche e pertanto fa parte del product sub-package. Il collegamento fra le features e i workingsteps rappresenta un collegamento fra il Process sub-package ed il Product sub-package.

La classe *MACHINING OPERATION* contiene tutte le informazioni riguardanti il contenuto tecnologico di un workingstep di lavorazione, cioè il percorso utensile, l'orientamento dello stesso, il piano, il punto di partenza del processo e le funzioni della macchina relative all'operazione, come il sistema refrigerante o quello di rimozione del truciolo, ecc. Inoltre, per la feature a cui l'operazione si riferisce, deve essere specificato l'utensile da utilizzare durante l'operazione ed i parametri tecnologici della lavorazione, quali la velocità del mandrino, l'avanzamento, ecc. Questi dati costituiscono parte integrante dell'operazione e non possono essere cambiati durante la relativa realizzazione. Da notare inoltre che l'operazione di lavorazione è direttamente associata all'utensile da usare. Questo collegamento rappresenta pertanto una relazione attiva fra il Process sub-package ed il System sub-package.

La classe *MACHINING WORKINGSTEP* è composta di una feature e di un'operazione di lavorazione. Si noti che la feature così come un'operazione di lavorazione può fare parte di più di un machining workingstep, cioè questa associazione non rappresenta un rapporto di uno a uno. La classe contiene l'indicazione della feature e dell'operazione di lavorazione che raggruppa insieme. Inoltre, contiene la descrizione dell'effetto finale sul workingstep, cioè il cambiamento nella geometria del pezzo effettuato in seguito alla lavorazione, come pure il security plain per il workingstep, per impedire eventuali collisioni.

I machining workingsteps sono dei building blocks che non devono essere eseguiti necessariamente in sequenza. Soltanto il workplan, modellato dalla classe *WORKPLAN*, determina l'ordine finale di realizzazione dei workingsteps. Di conseguenza, la lavorazione delle parti non si verificherà tipicamente ponendo i workingsteps in ordine sequenziale predeterminato, ma piuttosto secondo determinati criteri tecnologici come ad esempio la minimizzazione del numero di cambi utensile. Il workplan è definito come un insieme di workingsteps con una relativa sequenza di esecuzione. Un esempio di workplan codificato è il classico part program ISO. Poiché i workingsteps sono dei building blocks, possono essere organizzati in una framework a rete, considerando i vincoli di precedenza tecnologica tra loro. Questa proprietà consente di strutturare un dato numero di workingsteps in un cosiddetto NPP (Network Part Program). Un progetto nazionale italiano chiamato "Net.P.P. - Sviluppo del Network Part Program e sua introduzione nella lavorazione di parti prismatiche mediante asportazione di truciolo" (Development of the Network Part Program and its introduction in machining prismatic workpieces), è stato finanziato ed ha iniziato a studiare la struttura del NPP e gli effetti della relativa implementazione nel contesto produttivo. Di conseguenza, la classe *WORKPLAN* può essere sostituita, se necessario in futuro con la classe *NPP*.

Sia *MACHINING WORKINGSTEP* che *WORKPLAN* sono collegati alla classe *SETUP* del System sub-package. Un setup indica una configurazione particolare del prodotto, posizionato sul pallet tramite l'uso di un'attrezzatura che permette al sistema di realizzare un particolare set di workingsteps. Per questo motivo, un setup aggrega un insieme di workingsteps. Inoltre, ogni machining workingstep può essere associato ad uno o più setup. Un setup inoltre è associato ad un workplan. Questa relazione è implicitamente presente nel diagramma attraverso l'associazione di aggregazione fra la classe *SETUP* e la classe *MACHINING WORKINGSTEP*. Tuttavia, la scelta di rendere questa relazione esplicita è dovuta al fatto che a volte, nella fase di configurazione di sistemi produttivi, è utile considerare un workplan come una sequenza ordinata di setups anziché una sequenza di workingsteps. Infatti, in questo caso, è possibile stimare direttamente il numero di setups per la realizzazione del workplan ed il tempo necessario per completare il setup. Poiché il punto di vista adottato per sviluppare il modello è principalmente il punto di vista del sistema produttivo, è sembrato opportuno includere questo rapporto diretto. Le relazioni descritte collegano insieme il Product sub-package e il Process sub-package. Seguendo la stessa logica usata per il Product sub-package, anche

per il Process sub-package trattato si distingue il processo “as-designed”, dal processo “as-realized”. Per modellare quest’ultimo concetto, si introduce la classe *PROCESS*. Questa classe contiene tutte le informazioni di processo relative alla specifica lavorazione. La necessità di un simile tool di modellazione sarà più facile da capire dopo l’introduzione della classe *TRANSFORMATION*. La classe *PROCESS* contiene tutte le informazioni misurate relative alla realizzazione di un particolare tipo di processo. Un processo è quindi direttamente associato ad una o più trasformazioni. Inoltre, poiché il modello vuol essere generale ed applicabile alle varie realtà produttive, la classe *PROCESS* è stata specificata coerentemente con il tipo di processo specifico realizzato. Per formalizzare la process area del diagramma, sono state definite sei principali categorie: Manufacturing, Assembly, Disassembly, Transportation, Measurement and Washing.

Una classificazione del genere (figura 1) segue dalla formalizzazione proposta da Alting [35], che considera tre categorie di processi produttivi: mass conserving process, joining process e mass reducing process. I processi che conservano la massa (mass conserving processes) possono essere divisi in lavorazioni che coinvolgono materiali liquidi (es. fonderia), granulari (come ad esempio la metallurgia della polvere) o solidi (es. estrusione). Tipici joining processes sono processi di giunzione come la saldatura. I mass reducing process si riferiscono infine ad alcuni dei principali processi produttivi quali la fresatura, la tornitura, la foratura, la rettifica, ecc. ed anche a processi atipici quali il waterjet, il waterjet abrasivo, il laser e l’EDM (Electrical Discharge Machining). La classe dei mass reducing process è collegato con la classe *MACHINING OPERATION* che deriva da STEP-NC (ISO 14649), perché questi processi possono essere formalizzati usando lo standard STEP-NC, come descritto sopra. In particolare la fresatura (ISO 14649 Part 11/111), la tornitura (ISO 14649 Part 12/112) e l’EDM (ISO 14649 Part 13 and Part 14) sono già stati ampiamente descritti.

#### **5.4. Management sub-package (R1.2)**

Il Management sub-package è stato aggiunto al framework concettuale per la modellazione della struttura decisionale di un contesto produttivo, che prende le decisioni basandosi sull’osservazione dello stato dell’intero ambiente produttivo e sulla base di una strategia generale. Le difficoltà principali nella modellazione della struttura gestionale riguardano la formalizzazione delle diverse tipologie di decisioni e delle azioni di monitoraggio, a partire dal controllo del flusso delle parti e arrivando al singolo processo da controllare. Per modellare l’organizzazione di un sottosistema di management, si sono introdotte quattro classi aggiuntive. La classe principale a questo livello è la classe *MANAGER*. Questa classe può essere stanziata in maniera ricorsiva (come la classe *SYSTEM* del system sub-package) e rappresenta i decisori che controllano il comportamento dell’ambiente di produzione.

La possibilità di generare una gerarchia nella struttura dei responsabili è data dall’associazione di aggregazione che collega la classe con se stessa. Infatti, è frequente che nei contesti produttivi reali, sia presente più di un livello gerarchico dei responsabili, caratterizzato da livelli differenti di responsabilità di controllo. Oltre alla struttura gerarchica della classe *MANAGER*, ogni responsabile di più alto livello nella gerarchia è associato ad alcuni di livello più basso attraverso un’associazione chiamata “controls”. Inoltre, per collegare il Management sub-package agli altri sub-packages nel framework, il manager è collegato ad una serie di trasformazioni di cui è responsabile attraverso un’associazione chiamata “controls”. Quindi, il manager è dedicato al controllo sia di una o più trasformazioni, sia dei responsabili di livello più basso.

La classe *MANAGER* contiene informazioni riguardo quali trasformazioni e quali sub-managers si devono controllare, informazioni riguardanti lo schema di controllo, come ad esempio i parametri di controllo, i limiti di controllo, i percorsi nominali, le registrazioni, gli allarmi da generare ecc. ed informazioni riguardanti il tipo di parametri e di azioni su cui prendere le decisioni. Ogni responsabile per prendere le decisioni è supportato e guidato da una strategia. La strategia è modellata attraverso l’introduzione della classe *STRATEGY*. Il collegamento logico fra la strategia ed ogni livello della gerarchia decisionale è rappresentato con l’associazione “applies”, cioè ogni responsabile applica una strategia. La strategia è un’aggregazione di una serie di regole e di obiettivi che forniscono un insieme di azioni che possono essere intraprese dal decisore. Gli obiettivi della strategia sono modellati dalla classe *OBJECTIVE*. Questa classe contiene le informazioni riguardanti le motivazioni dell’azione di controllo ad un particolare livello di aggregazione. Queste possono essere per esempio, ad un basso livello, di ridurre i costi o di aumentare la velocità di una trasformazione spaziale, o, ad alto livello, per esempio di ridurre il time to market o di ridurre l’impatto ambientale. La classe *RULE* infine include l’insieme delle regole di management

disponibili affinché la strategia raggiunga gli obiettivi definiti. Quindi, la classe *STRATEGY* contiene un insieme di obiettivi e di regole che possono essere usate dal responsabile per realizzare la funzione di management.

### 5.5. Integrazione: La classe Transformation

L'ultima classe trattata in questa sezione è la classe *TRANSFORMATION*. Questa svolge un ruolo fondamentale nel framework poiché di per se non fa parte di nessun sub-package ma piuttosto rappresenta l'integrazione fra i quattro sub-packages definiti. In particolare, la trasformazione esprime le interazioni che riguardano il sistema mentre realizza un dato processo su un particolare prodotto. Di nuovo, le trasformazioni possono essere di specie o spaziali. Le trasformazioni di specie sono applicate nel caso in cui cambiano le proprietà geometriche o fisiche del prodotto. Le trasformazioni spaziali sono applicate nel caso si verifichi una modifica della posizione dei prodotti nel sistema produttivo. Essendo il sistema composto da un insieme di risorse fisiche, esso può essere oggetto della trasformazione stessa. Questo è il caso in cui, per esempio, alcune risorse, tipicamente i buffer, sono spostati da una posizione all'altra. In questo caso, nessun prodotto è coinvolto nella trasformazione ma le risorse del sistema sono esse stesse i prodotti dell'applicazione di un dato processo. Questa è la ragione per la quale si è consentito al modello la possibilità che il prodotto non sia coinvolto in una trasformazione, attribuendo all'associazione fra *TRANSFORMATION* e *PRODUCT* molteplicità uguale a zero dal lato del prodotto.

Oltre agli oggetti fisici, anche i decisori sono coinvolti nella trasformazione e pertanto collegati con questa classe. In particolare, si assume che le azioni di controllo e gestione intraprese dai responsabili nell'ambiente produttivo siano applicate nel momento in cui avviene la trasformazione. Questa ipotesi è il risultato di diversi casi di applicazione delle azioni di management.

Anche le azioni di controllo del flusso produttivo sono applicate nel momento in cui un prodotto è processato da una trasformazione spaziale. Per quanto riguarda le azioni di controllo del processo produttivo, queste impattano sul comportamento produttivo nel momento in cui avviene una trasformazione di specie. Le azioni di cambio utensile sono invece effettuate dopo che una trasformazione di specie sia finita o prima che una trasformazione di specie cominci. Di conseguenza, le decisioni più frequenti che sono prese per gestire un contesto produttivo, hanno effetto una volta avvenuta una certa trasformazione. Può essere quindi definitivamente ipotizzato che i manager controllino la realizzazione delle trasformazioni.

La classe *TRANSFORMATION* contiene le informazioni per quanto riguarda il tempo dell'occorrenza della trasformazione globale, la durata della stessa, la lista dei sistemi coinvolti, il prodotto fisico in questione, i processi fisici ed i manager addetti all'azione di controllo, oltre alla lista delle azioni di controllo applicate. Ricapitolando e concludendo, ogni volta che viene modellata una trasformazione, si verifica un'integrazione fra prodotto, processo ed sistema. Questo è esattamente quello che si vuole rappresentare con il concetto di integrazione nel framework di riferimento.

## 6. Prime applicazioni

### 6.1. Linea flessibile che produce piastre portanti per alberi a camme

In quanto segue, si presenta l'applicazione del reference framework per modellare un sistema produttivo reale. In particolare, sarà presentato come gli elementi di modellazione che appartengono alla System Area possono essere utilizzati per rappresentare il sistema reale con livelli differenti di dettaglio.

Il sistema reale (figura 3) è una linea transfer flessibile che produce piastre portanti per alberi a camme di testate di motori 4/5 cilindri. Il termine "flessibile" è usato per indicare che ogni stazione della linea transfer si compone di un dato numero di macchine utensili flessibili, in particolare centri di lavoro a 5 assi. In totale, ci sono tre stazioni, indicate in Figura 3 dai nomi OP 10, OP 20 ed OP 30 e tre buffer inter-operazionali, chiamati Buffer 1, Buffer 2 e Buffer 3. Ciascuna operazione è composta di tre centri di lavoro identici ed ogni centro di lavoro è dotato di un dispositivo APC (Automatic Pallet Changer), con capacità di stoccaggio di due pallet. Ogni stazione è inoltre dotata di un sistema di handling per il trasporto del pezzo in lavorazione dal buffer upstream al centro di lavoro e dal centro di lavorazione al buffer downstream. Ogni sistema di material handling movimentava direttamente il pezzo in lavorazione, senza il bisogno di nessun dispositivo pallet.

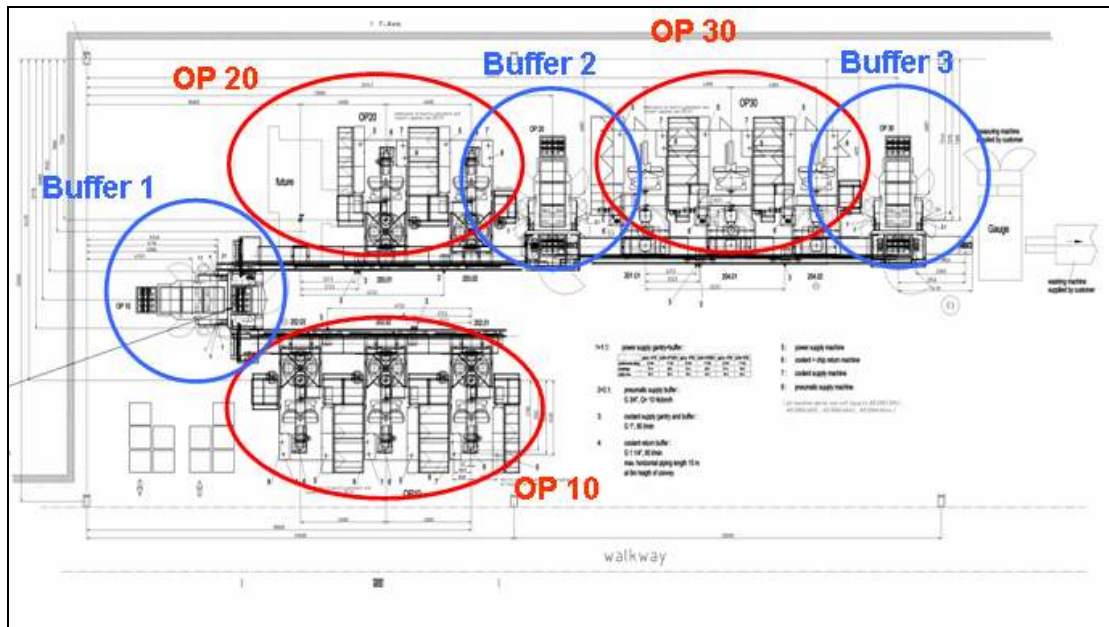


Fig. 3 Linea flessibile per la produzione di piastre portanti per alberi a camme

Il flusso di materiale si comporta come segue. I pezzi in lavorazione entrano dalla parte posteriore del Buffer 1 (da sinistra di figura 3) ed inizialmente vengono stoccati nel buffer. Poi, uno alla volta, i workpieces sono portati all'OP 10 per essere caricati su una posizione libera dell'APC di uno dei centri di lavorazione, in cui saranno successivamente lavorati. Il trasporto è effettuato da un dispositivo di presa. Nel caso in cui nessuno dei dispositivi APC della stazione OP20 sia in grado di prendere in consegna il pezzo, lo stesso dispositivo riporta il pezzo dal centro di lavoro al Buffer 1, dov'è movimentato da un dispositivo di presa appartenente al Buffer 1 o, in caso contrario, direttamente all'OP 20. A valle della stazione OP 20, il flusso di materiale ripete sostanzialmente quanto descritto per la stazione OP 10, va inizialmente verso il Buffer 2, poi verso la stazione OP 30 ed infine raggiunge il Buffer 3.

La stazione OP 10 è principalmente riservata ad operazioni di fresatura, oltre ad alcune di foratura. Nella stazione OP 20 si realizza un set di operazioni di foratura e la maggior parte di quelle di filettatura. Infine, disporre l'OP 30 è riservata per le ultime operazioni di alesatura e filettatura

Ora, si immagini che si voglia analizzare il comportamento dinamico di un tale sistema, mantenendo l'approccio il più semplice possibile. Con riferimento al modello presentato nella sezione precedente, si può descrivere (figura 4) la linea di produzione sia come un sistema composto dai seguenti sottosistemi: OP 10, OP 20, OP 30, Buffer 1, Buffer 2 e Buffer 3, sia come un'associazione di aggregazione fra la classe *SYSTEM* del class diagram di riferimento e se stesso. Ogni sottosistema è così di per se un sistema, nel senso che è un sistema complesso che può avere sia un insieme di sottosistemi sia un set di risorse. Se non sono necessari, i dettagli relativi a questi sottosistemi possono essere trascurati e l'analisi può essere interrotta al livello precedente, solo definendo gli attributi generali come la capacità dei buffers, o il tempo di servizio ed il numero di macchine di ogni stazione. Gli attributi di base come questi possono infatti essere sufficienti, in taluni casi.



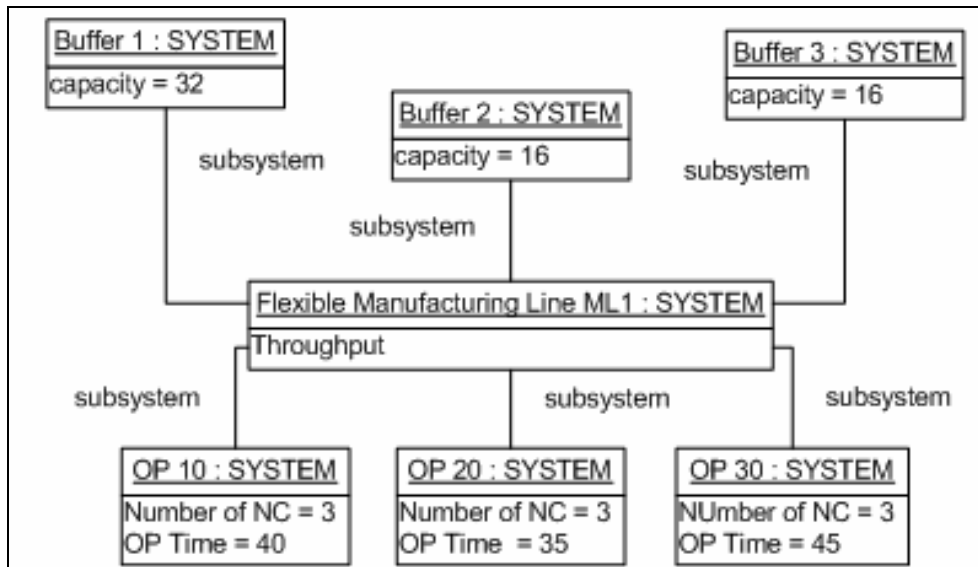


Fig. 4 Diagramma ad oggetti della linea reale flessibile

Nel caso in cui lo scopo dell'analisi richieda informazioni più dettagliate riguardo i sottosistemi ed i componenti delle operazioni e dei buffers, allora gli oggetti e gli attributi di figura 4 non sono più sufficienti.

Nell'ipotesi che ulteriori informazioni siano disponibili, come ad esempio quelle presentate in figura 5, allora può essere sviluppato un modello più dettagliato della linea di produzione.

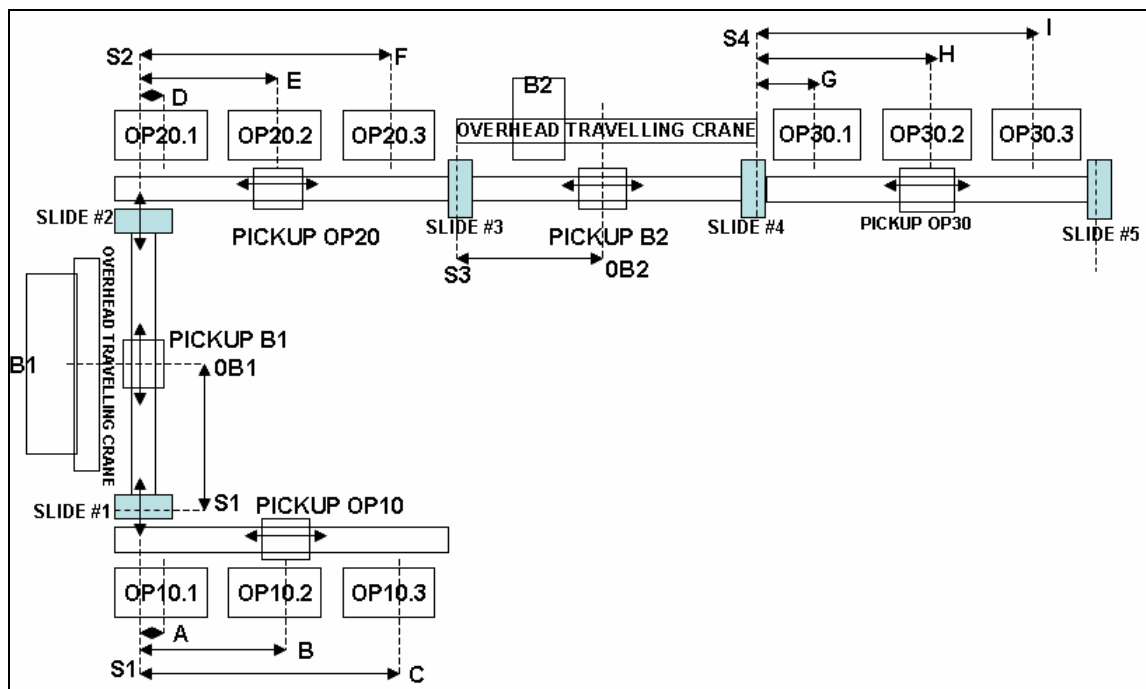


Fig. 5 Specificazione dettagliata del sistema composto da operazioni e buffers

Per quanto riguarda figura 5, il Buffer B1 è composto nella specifico dal buffer stesso e, dalla gru mobile sovrastante, due scorrevoli e il sistema di presa citato in precedenza. Tutti questi oggetti sono collegati al sottosistema Buffer B1 per mezzo dell'associazione di composizione, come dal framework di riferimento.

Gli attributi in questo caso sono orientati più al basso livello come ad esempio la velocità del sistema di presa o la velocità di ogni scorrevole.

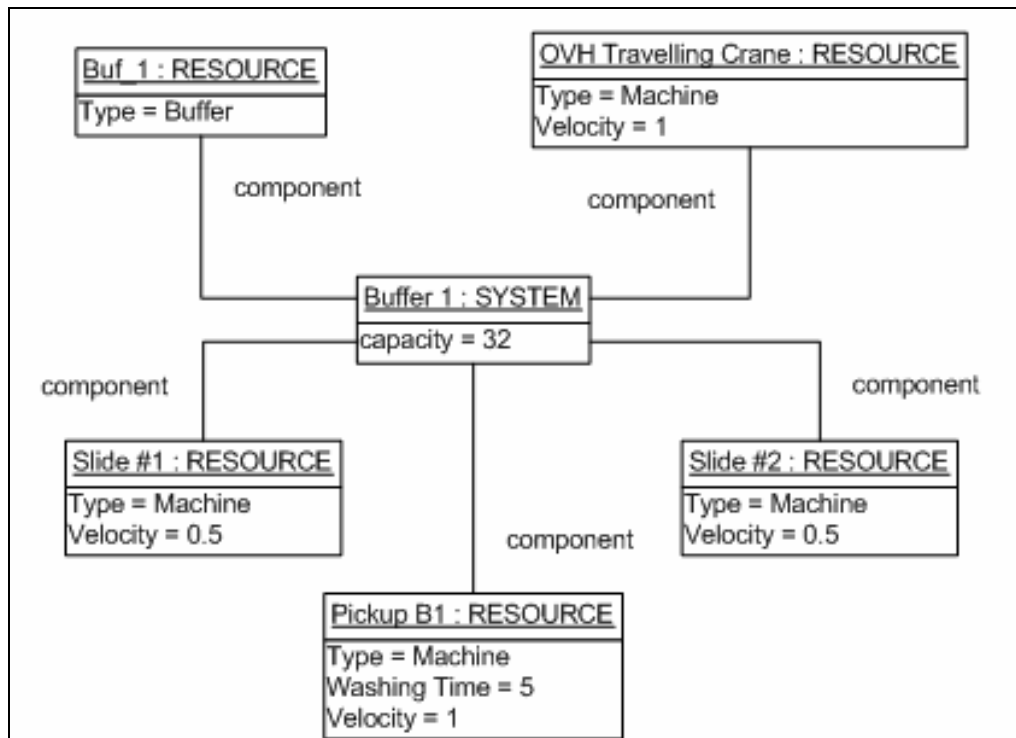


Fig. 6 Object diagram del Buffer 1

La scalabilità del modello, come introdotto nella sezione 1 e dettagliata nella sezione 5 è perciò dimostrata già nel caso di questa linea di produzione flessibile. Questa scalabilità è stata sfruttata per sviluppare i template object oriented della simulazione per realizzare la simulazione ad eventi discreti del sistema reale a diversi livelli di dettaglio.

## 6.2. Il Problema di Configurazione del Sistema Produttivo

In questo paragrafo è presentata l'applicazione del framework generale di riferimento al problema di configurazione di sistemi produttivi. Essa mira a descrivere i sistemi produttivi che realizzano operazioni di lavorazione; in particolare l'attenzione è focalizzata sugli Automated Manufacturing Systems che consistono in macchine utensili NC con dispositivi automatici di material handling (come ad esempio FMS con macchine parallele, FMS con layout multi-cella e composti di celle non identiche o Focused Flexibility Manufacturing System). La seguente formalizzazione dei dati può essere utile per un progettista di sistema quando deve risolvere il problema di configurazione o riconfigurazione di sistema.

La fase di configurazione di sistema svolge un ruolo fondamentale per le prestazioni generali dei sistemi produttivi che devono affrontare il trade-off tra produttività e flessibilità. Il problema consiste nella progettazione della configurazione di sistema ottimale, cioè il numero e la tipologia di risorse necessarie per soddisfare correttamente i requisiti. I requisiti tecnologici dei prodotti che devono essere realizzati porta alla scelta delle tipologie di risorse da configurare, mentre i requisiti in termini di volumi produttivi guidano la scelta riguardo il numero di risorse da configurare. La configurazione di sistema è altamente critica dal punto di vista industriale, a causa delle numerose questioni economiche e tecnologiche che devono essere considerate. Il processo può essere costoso e richiede molto tempo, poiché devono essere considerati molti aspetti qualitativi e quantitativi; alcune di queste decisioni devono essere prese coinvolgendo il top management dell'azienda e questo comporta costi elevati, così evidenziando la necessità di strumenti che rendano la procedura di configurazione del sistema più efficiente, riducendone il tempo trattato e più efficace, aumento la probabilità di scegliere la configurazione migliore. Uno strumento di supporto potrebbe contribuire nell'esplorare più di una soluzione possibile, così fornendo un vantaggio

competitivo nel caso di problemi complessi, che sarebbero in caso contrario descritti superficialmente ed affrontati con un approccio non formalizzato.

Uno strumento di supporto efficace ha bisogno, per funzionare correttamente, di una formalizzazione completa e precisa dei dati. Il framework di riferimento descritto nella parte 3 è stato applicato proprio per questa finalità. In particolare, l'attenzione è stata focalizzata sulla tipologia architetturale del sistema di produzione flessibile (FMS) in entrambi i casi di cella singola e multipla. Quindi, le risorse del sistema considerate sono macchine, trasportatori, stazioni di carico/scarico e pallets. Al momento, risorse più dettagliate come le attrezzature non sono considerate, poiché si suppone che le decisioni riguardanti queste risorse siano prese ad un livello più basso nella gerarchia di configurazione di sistema.

Figura 7 presenta un class diagram UML con la struttura di dati concettuale sviluppata per il problema di configurazione di sistemi produttivi. Molte delle classi appartenenti al framework di riferimento possono essere riutilizzate anche in questo caso specifico. Le differenze principali riguardano la mancanza di classi che appartengono alla Management Area ed il basso livello di dettaglio nella descrizione delle informazioni di processo.

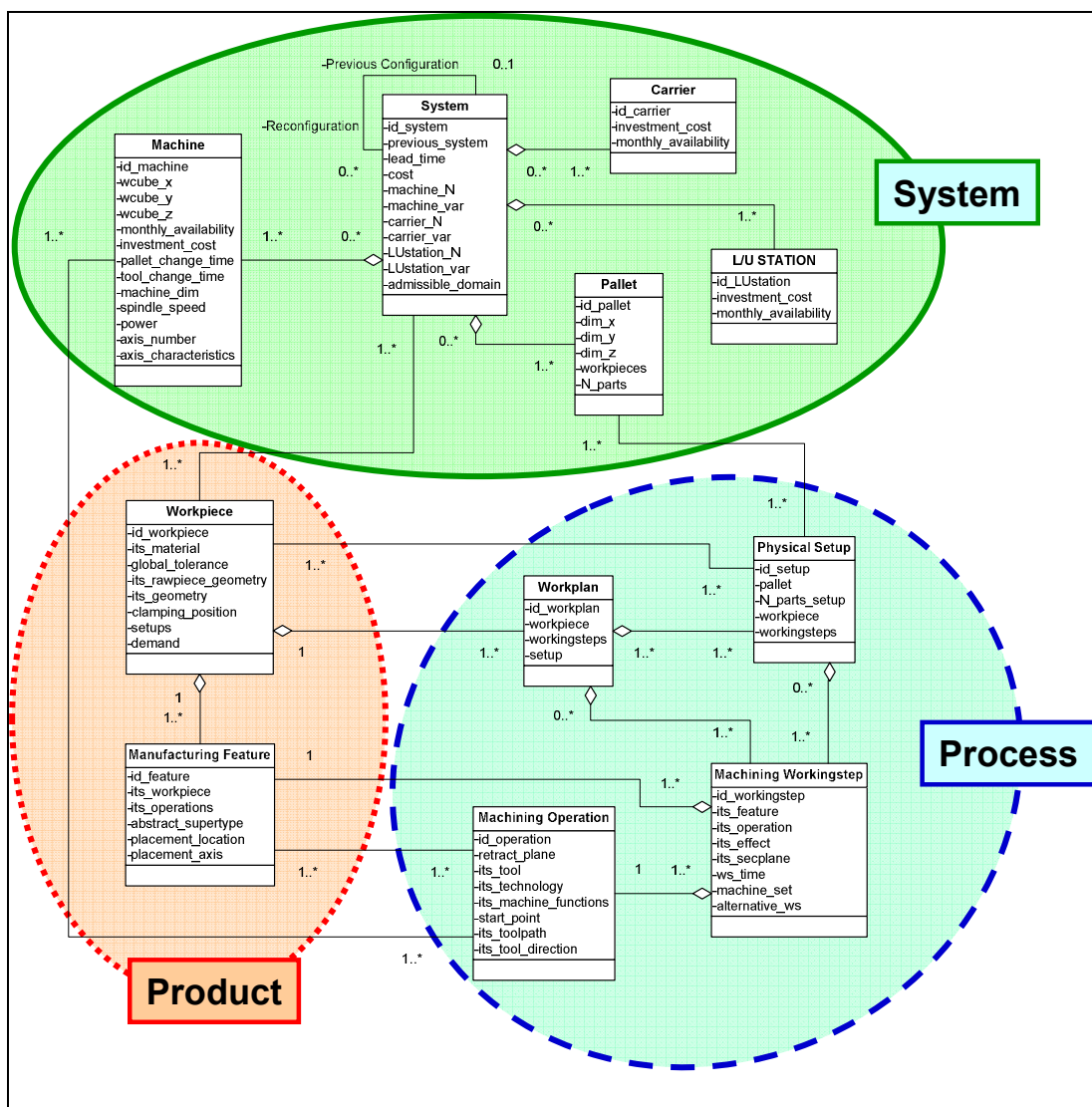


Fig. 7 Class diagram UML per il problema di configurazione di sistemi produttivi

Analizzando il class diagram di figura 7, è possibile identificare tre aree principali, evidenziate con gli ovali di diverso colore: le aree sistema, prodotto e processo.

Nella System Area (ovale verde/linea continua) sono dettagliate le caratteristiche architettoniche del sistema produttivo. La classe System (si riferisca alla tabella 1: Dettaglio degli attributi della classe SYSTEM) è particolarmente cruciale, poiché la definizione della configurazione del sistema è l'obiettivo finale di tutto il processo di formalizzazione dei dati e del problema stesso di configurazione. La dinamica della configurazione del sistema produttivo può anche essere rappresentata, grazie all'attributo "previous\_system", che collega un dato sistema con la sua precedente configurazione. L'attributo "admissible\_domain" definisce le capabilities del sistema, cioè le possibili combinazioni di volumi produttivi relativi ai diversi workpieces che il sistema può produrre.

Il sistema in se è composto dai relativi elementi fisici: le macchine, i trasportatori, le stazioni di carico/scarico ed i pallets. Le istanze (cioè gli oggetti) di queste classi sono sia le risorse attualmente parte della configurazione corrente del sistema, sia quelle disponibili nel catalogo del progettista. I loro attributi definiscono le caratteristiche tecnologiche, fisiche e di costo di queste risorse.

<b>Attributi della classe SYSTEM</b>	
Id_system	Identificatore
previous_system	Identificatore della precedente configurazione di sistema
lead_time	Lead time dall'emissione dell'ordine alla completa installazione del nuovo sistema configurato
Cost	Costi d'investimento relativi alla nuova configurazione di sistema (Offerta commerciale del System Designer)
machine_N	Numero di ogni tipo di macchina nel sistema
machine_var	Variazione nel numero di ogni tipo di macchina rispetto alla configurazione precedente
carrier_N	Numero di ogni tipo di carrier nel sistema
carrier_var	Variazione nel numero di ogni tipo di carrier rispetto alla configurazione precedente
LUstation_N	Numero di ogni tipo di stazione di carico/scarico nel sistema
LUstation_var	Variazione nel numero di ogni tipo di stazione di C/S rispetto alla configurazione precedente
admissible_domain	Coefficienti che definiscono il dominio ammissibile della configurazione di sistema cioè il mix produttivo che il sistema sarebbe in grado di produrre

*Tab. 1 Dettaglio degli attributi della classe SYSTEM*

La "Product Area" Linea (ovale rossa/linea punteggiata) consiste nelle classi workpiece e Machining feature; entrambe le classi derivano dallo standard STEP-NC (ISO14649). Ogni oggetto della classe workpiece è uno dei tipi di prodotto realizzato dal sistema ed è in rapporto uno a uno con il codice del prodotto come nel caso degli ordini da cliente. Rispetto alla classe workpiece di STEP-NC, la classe attuale è arricchita con l'attributo "demand" che aggiunge le informazioni riguardanti la domanda futura prevista per la parte.

La Process Area descrive come le risorse del sistema possono essere usate per lavorare i pezzi in lavorazione. Anche le classi Machinig Operation, Machining Workingstep e Workplan derivano dallo standard STEP-NC (ISO14649).

Gli oggetti della classe Machinig Operation descrivono il processo di lavorazione per un'area limitata del pezzo, specificando, come minimo, l'utensile da utilizzare e un insieme di parametri tecnologici. Gli oggetti della classe Machinig Workingstep rappresentano invece il processo di lavorazione per un'area specificata del pezzo che non può esistere indipendentemente da una feature. I machining workingsteps specificano l'associazione fra una determinata feature e l'operazione necessaria per realizzare quella

feature. Un Workplan è semplicemente una raccolta di Workingsteps insieme ad una sequenza esecutiva che deve essere rispettata per realizzazione quel determinato set di Workingsteps.

Rispetto allo standard STEP-NC, la classe Workingstep qui mostra inoltre gli attributi “ws\_time”, “machine\_set” e “alternative\_ws”. L’attributo “ws\_time” modella il tempo macchina necessario per effettuare il workingstep, l’attributo “machine\_set” rappresenta l’insieme dei tipi di macchina che possono processare quel workingstep ed infine l’attributo “alternative\_ws” specifica l’insieme dei workingsteps che possono essere scelti come alternativa al workingstep attuale.

L’implementazione del modello di formalizzazione descritto in precedenza è un aspetto chiave da considerare, poiché il framework proposto mira ad essere applicato a casi reali. Questa operazione fa parte delle attività del progetto nazionale italiano chiamato “Methodologies and Tools for the configuration of production systems with focused flexibility”, fondato dal “MiUR – Ministero dell’Università e della Ricerca” nel quadro “PRIN - Programmi di Ricerca di Interesse Nazionale”. Questo genere di conoscenza può essere rappresentato sia attraverso un database sia attraverso un’ontologia. Un database può dare una visione più concreta e più specifica del mondo, mentre un’ontologia è usata per generare un modello concettuale del mondo; un database si focalizza sulle istanze, mentre un’ontologia sulle entità. Inoltre, un’ontologia può essere analizzata con i metodi “che ragionano” che possono contribuire ad estendere la conoscenza.

Durante il lavoro, sono state sviluppate entrambe le implementazioni; la soluzione migliore dipende dalla particolare applicazione e verrà ulteriormente analizzata nelle seguenti attività del progetto. Il database relazionale è stato implementato utilizzando MS Access, mentre l’ontologia è stata sviluppata usando Protégé (<http://protege.stanford.edu/>).

Le tabelle e le relazioni del database sono riportati in figura 8 e 9, mentre le classi dell’ontologia e gli attributi “System class” sono mostrati in figura 10. Si può vedere che l’implementazione del database relazionale richiede un gran numero di relazioni ed un elevato numero di entità rispetto all’ontologia; infatti un’ontologia permette una più semplice implementazione delle relazioni, in particolare la relazione “IS-A”.

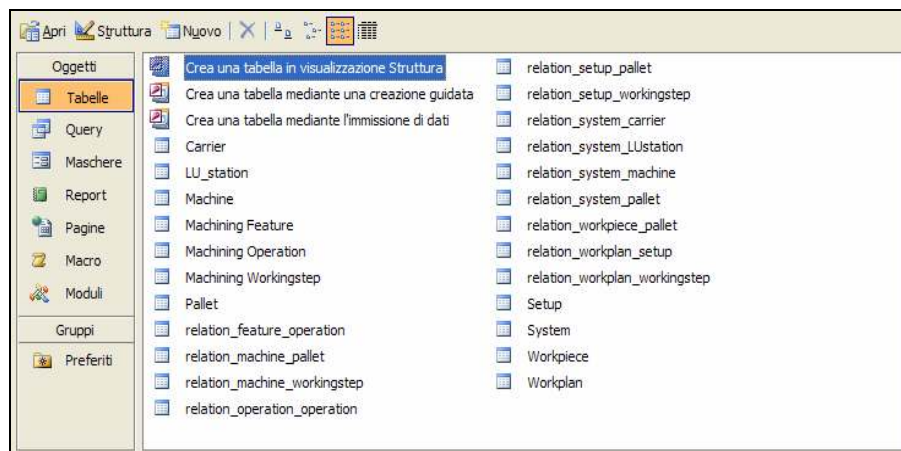


Fig. 8 Implementazione in MS Access (1)

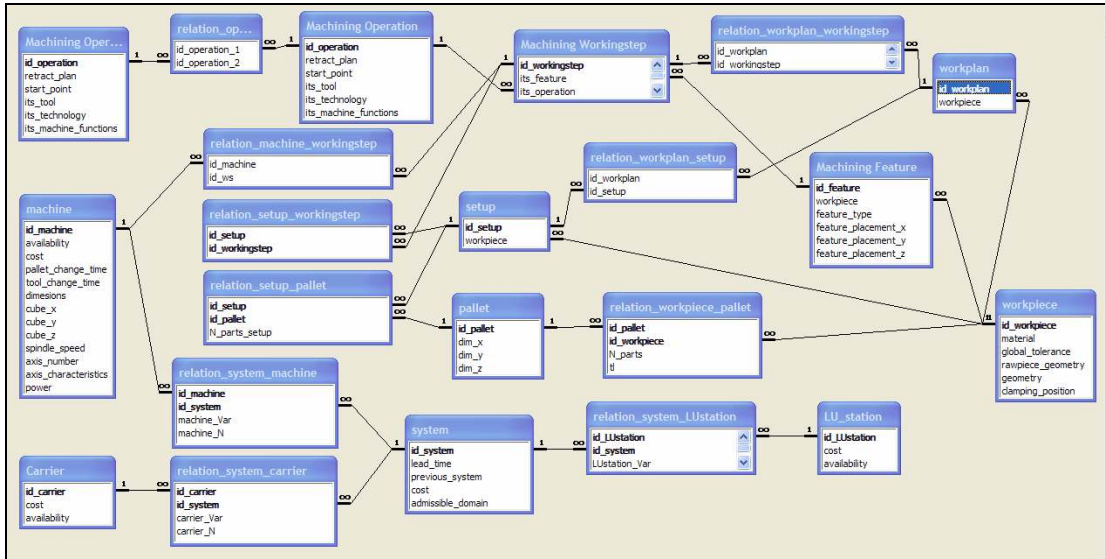


Fig. 9 Implementazione in MS Access (2)

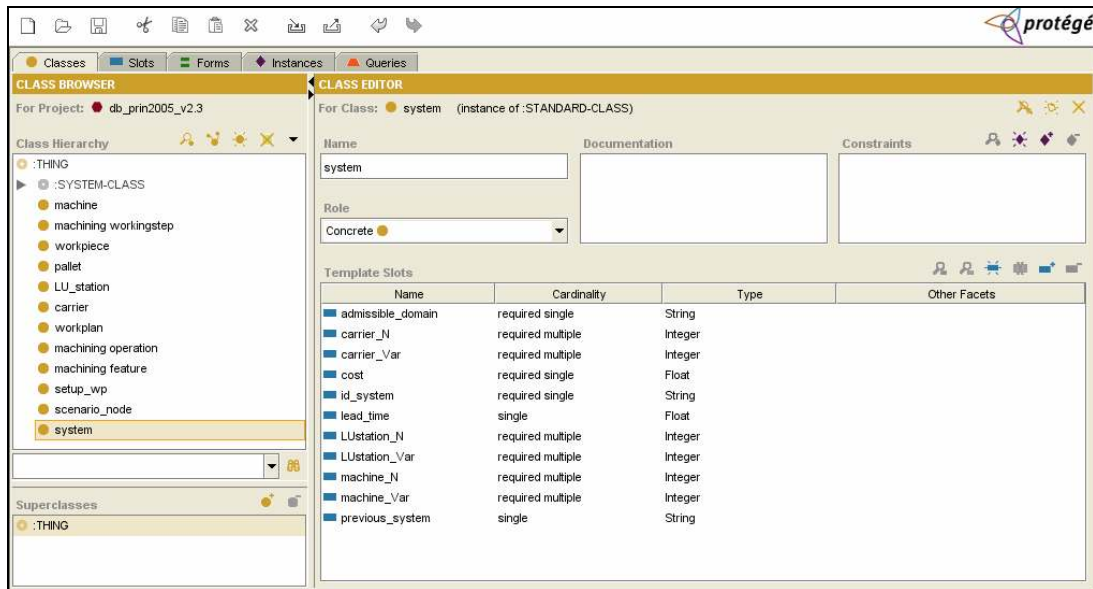


Fig. 10 Implementazione in Protégé



- [3] Y. Dallery, S.B. Gershwin, "Manufacturing Flow Line Systems: A Review of Models and Analytical Results, Queueing Systems Theory and Applications", Special Issue on Queueing Models of Manufacturing Systems, 12(1-2) , 3-94, 1992.
- [4] M. Colledani, T. Tolio, "A Decomposition Method to Support the Configuration/Reconfiguration of Production Systems", Annals of CIRP, 54/1, 441-445, 2005.
- [5] H. Bley, C.-C. Wuttke, "Distributed Simulation Applied to Production Systems", Annals of the CIRP, 46/1: 361-364, 1997.
- [6] S. Toma, T. Tolio, Q. Semeraro, A. Grieco "Simulation of tool and part flow in FMSs", International Journal of Production Research vol 33, pg. 643-658, 1995.
- [7] L.C. Alwan, "Statistical Process Analysis", McGraw-Hill, 2000.
- [8] D.C. Montgomery, "Design and Analysis of Experiments", Wiley, 2000.
- [9] F. Casati, B. Pernici, "Linguaggi per la modellazione dei processi aziendali", <http://www-db.deis.unibo.it/courses/SIG/processi.pdf>
- [10] A. Matta, T. Tolio, M. Tomasella, P. Zanchi, "A detailed UML model for general flexible manufacturing systems", Proceedings of the 4th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering CIRP ICME '04. June 30- July 2, 2004, Sorrento, Italy. pp. 113-118.
- [11] M. Bruccoleri, S. Noto La Diega, G. Perrone, "An Object-Oriented approach for flexible manufacturing system control system analysis and design using unified modeling language", The Int. J. of Flexible Manufacturing Systems, 15, 195-216, 2003.
- [12] T.Y. Park, K.H. Han, B.K. Choi, "An object-oriented modelling framework for automated manufacturing system", International Journal of Computer Integrated Manufacturing, 10/5, 324-334, 1997.
- [13] P. Kellert, N. Tchernev, C. Force, "Object-oriented methodology for FMS modeling and simulation", International Journal of Computer Integrated Manufacturing, 10, 404-434, 1997.
- [14] C.-M. Liu, C.-F. Chien, I.-Y. Ho, "An object-oriented analysis and for shop floor control systems design method", International Journal of Computer Integrated Manufacturing, Volume 11, Number 5, 1 September 1998, pp. 379-400(22)
- [15] G. Booch, I. Jacobson, J. Rumbaugh, *The Unified Modelling Language Reference Manual, Second Edition*, The Addison-Wesley Object Technology Series, , 2004.
- [16] H. Van Brussel, P. Valckenaers, L. Bongaerts, J. Wyns, "Architectural and System Design issues in Holonic Manufacturing Systems", Pre-prints of IMS'95, IFAC workshop, Bucharest, Romania, October 1995.
- [17] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, P. Peeters, "Reference Architecture for Holonic Manufacturing Systems: PROSA", Computers In Industry, special issue on intelligent manufacturing systems, Vol. 37, No. 3, pp. 255 - 276, 1998.
- [18] S. Narayanan, D.A. Bodner, U. Srekanth, T. Govindaraj, L.F. McGinnis, C.M. Mitchell, "Research in object-oriented manufacturing simulations: An assessment of the state of the art", IIE Transactions, 30(9), 795-810, 1998.
- [19] [http://www.species.polimi.it/downloads/SPECIES\\_MCM.pdf](http://www.species.polimi.it/downloads/SPECIES_MCM.pdf)



- [20] F.L. Krause, F. Kimura, T. Kjellberg, S. C.-Y. Lu, "Product Modelling", Annals of the CIRP Vol. 42/2/1993.
- [21] A. Bufardi, D. Kiritsis, P. Xirouchakis, "Research issues on product lifecycle management and information tracking using smart embedded systems", Advanced Engineering Informatics, Vol.17/3-4/2003
- [22] Process Specification Language (PSL), <http://www.mel.nist.gov/psl/>
- [23] <http://www.step-nc.org/>
- [24] F. Kimura, "Product and Process Modelling as a Kernel for Virtual Manufacturing Environment", Annals of the CIRP Vol. 42/1/1993 pg. 147-150.
- [25] A. Thibault, A. Siadat, R. Bigot, P. Martin, "Method for Integrated Design Using a Knowledge Formalization", Proceedings of 3rd Conference on Digital Enterprise Technology, Setubal, Portugal, 18th-20th Sept. 2006.
- [26] O. López-Ortega, R. Moramay, "A STEP-based manufacturing information system to share flexible manufacturing resources data", Journal of Intelligent Manufacturing, Volume 16, Number 3, pp.287-301, 2005.
- [27] H.K. Lin, J.A. Harding, M. Shahbaz, "Manufacturing System Engineering Ontology for Semantic Interoperability Across Extended Project Teams", International Journal of Production Research , Vol 42 Number 24, pages 5099-5118, 2004
- [28] J.A. Harding, K. Popplewell, D. Cook, "A Manufacturing System Engineering Moderator: An Aid for Multi-discipline project teams", International Journal of Production Research, Vol 41 Number 9, pages 1973-1986, 2003.
- [29] J. Zhao, W.M. Cheung, R.I.M. Young, R. Bell, "An Object Oriented Manufacturing Data Model for a Global Enterprise", 15th International Conference on Computer-Aided Production Engineering (CAPE'99), Durham, UK, 19-21st April, 1999, pp.582-588
- [30] M.S. Fox, M. Gruninger, "On Ontologies and Enterprise Modelling", International Conference on Enterprise Integration Modelling Technology 97, Springer-Verlag, 1997
- [31] K. Kosanke, F. Vernadat, M. Zelm, "CIMOSA: enterprise engineering and integration", Computers in Industry, Volume 40, Number 2, pp. 83-97, 1999
- [32] J.U. Zimmermann, "Informational Integration of Product Development Software in the Automotive Industry – The ULEO Approach", Ph.D. Thesis, University of Twente, 2005.
- [33] J. Rumbaugh, "Object-Oriented Modeling and Design", Prentice Hall, 1991
- [34] M. Fowler, "UML Distilled: A brief guide to the Standard Object Modelling Language, Third edition", Addison-Wesley Professional, 2003.
- [35] L. Alting, "Manufacturing Engineering Processes", Marcel Dekker, Inc., 1994.