

---

# Samenvatting

## 1. Inleiding

### 1.1 Probleemstelling

Doorheen de geschiedenis is het succes van een samenleving dikwijls bepaald geweest door de efficiëntie en de effectiviteit waarmee ze haar schaarse productiemiddelen kon gebruiken. Productiebeheer is nooit een sinecure geweest, ook niet in de huidige industriële context, ook niet met de komst van de computer.

Een van de kritische aspecten van productiebeheer is het plannen en on line besturen van het productiesysteem. Het verschil in performantie tussen een goede en een slechte planning is dikwijls aanzienlijk en men heeft aangetoond dat de rekentijd nodig voor het berekenen van een optimale planning excessief hoog is voor problemen van een realistische grootte. Bovendien wordt de kwaliteit van een planning, zelfs al heeft ze een goede performantie, vaak nadelig beïnvloed door het optreden van storingen. Dit is een van de belangrijke redenen waarom de onderzoeksresultaten van de voorbije decennia op het gebied van fijnplanning niet of slechts bij mondjesmaat werden toegepast in industriële situaties.

De eerste implementaties van computer-geïntegreerde werkvloerbesturingssystemen zijn hiërarchisch ontwikkeld. Hiërarchische besturingssystemen zijn een belangrijke stap voorwaarts ten opzichte van gecentraliseerde besturingen, maar blijken traag te reageren op storingen. Een veelbelovend alternatief zijn de hiërarchie-loze besturingssystemen, ook wel heterarchische besturingen genoemd. Ze zijn gebaseerd op het mechanisme van een vrijemarkteconomie en hebben op dezelfde manier een adaptief, zelfregelend karakter. Ze nemen echter ook de onvoorspelbaarheid over en het is moeilijk om met heterarchische systemen de globale performantie te optimaliseren.

### 1.2 Doelstellingen

Het doel van deze thesis is de ontwikkeling van een productiebesturing die een hoge en betrouwbare performantie levert en toch goed en snel reageert op storingen. Het uitgangspunt daarvoor is dat dergelijk gedrag kan ontstaan door een goede combinatie van hiërarchische en heterarchische principes. Daarvoor moeten de nodige concepten geconcipieerd worden, die algemeen bruikbaar zijn over een brede waaier van productiesystemen en -omstandigheden. Daarnaast moet er echter ook een onderzoeksprototype gebouwd worden dat de uitgedachte concepten uitwerkt en toetst aan de realiteit. Dit alles omvat de uitwerking van een besturingsarchitectuur en het ontwerp van een aantal besturingsalgoritmen om het beoogde doel te bereiken. In de loop van het mijn doctoraatswerk werd het duidelijk dat het onderwerp van deze the-

sis een typisch toepassingsdomein is van het concept van holonische productiesystemen, zoals gedefinieerd door Suda (1989, 1990), en uitgewerkt in het IMS-programma<sup>1</sup> (Hayashi, 1993).

### 1.3 Verwezenlijkingen

De resultaten van deze thesis bestaan uit drie belangrijke besturingsconcepten, uitgewerkt in een architectuur en een aantal algoritmen. Concreet heeft dat geresulteerd in een prototype-implementatie, waarop experimenten met numerieke resultaten werden uitgevoerd.

Een eerste belangrijk besturingsconcept is de invoering van *hiërarchie in gedistribueerde systemen*. Dit concept resulteert in een besturingsarchitectuur die zowel hiërarchische als heterarchische besturing ondersteunt. Het basisprincipe van die architectuur is dat hij bestaat uit een tandemoplossing van een fijnplanningsholon en een on-line-werkvloerbesturingsholon.

Een tweede concept behelst het *verdelen van beslissingsbevoegdheid* over de verschillende agenten in het systeem. Dit concept blijkt krachtiger dan het beurtings toekennen van de volledige beslissingsbevoegdheid aan die agenten. Om dit te modelleren is een model voor de beslissingsbevoegdheidsverdeling uitgewerkt.

Ten derde stelt deze thesis het concept van *gelijktijdige fijnplanning en planningsuitvoering* voor. Dit concept laat toe om snel te reageren op storingen en toch de performantie te optimaliseren. De concrete realisatie hiervan is een samenwerkingsmodel op basis van Petrinetten. Uit dit model volgt dat niet alleen reactieveplanningsmethoden (zoals reeds beschreven in de literatuur — zie hoofdstuk 2) nodig zijn, maar ook on-line-besturingsalgoritmen om een planning uit te voeren. Deze thesis toont een aantal voorbeelden van dergelijke planningsuitvoeringsalgoritmen en toont aan — aan de hand van praktische implementaties en experimentele resultaten — dat het gebruik van dergelijke algoritmen een hogere performantie geeft aan de productiesystemen.

Deze thesis heeft ook concrete bijdragen geleverd aan het GOA-project van de afdeling PMA van de K.U.Leuven over holonische productiesystemen en aan het HMS-project in het internationale IMS-programma. De besturingsarchitectuur die ontwikkeld is voor deze thesis was een startpunt voor de ontwikkeling van een holonische referentiearchitectuur (Wyns, 1999). De algoritmen zijn voorbeelden die aangeven hoe de algemene concepten over holonische productiesystemen effectief kunnen toegepast worden.

### 1.4 Overzicht van de thesis

Deze thesis is gestructureerd als volgt. Het tweede hoofdstuk geeft een overzicht van de bestaande literatuur over fijnplanning, werkvloerbesturing en besturingsarchitecturen. Daarbij is zowel de meer traditionele hiërarchische aanpak als de recentere heterarchische aanpak behandeld. Het derde hoofdstuk schetst het kader van holonische productiesystemen en plaatst de resultaten van deze thesis in een breder kader.

---

<sup>1</sup> Intelligent Manufacturing Systems.

De echte resultaten worden beschreven vanaf hoofdstuk 4. Dat hoofdstuk beschrijft de ontwikkelde concepten en de besturingsarchitectuur die daaruit resulteerde. Hoofdstuk 5 modelleert de samenwerking tussen holons aan de hand van Petri-netten. Hoofdstuk 6 beschrijft de planningsuitvoeringsalgoritmen.

De drie laatste hoofdstukken sluiten de thesis af. Hoofdstuk 7 beschrijft de praktische implementaties voor deze thesis. Hoofdstuk 8 beschrijft de experimenten die uitgevoerd zijn om de verbeteringen van holonische systemen numeriek aan te tonen. Hoofdstuk 9 vat alle belangrijke besluiten nog eens samen.

## 2. Literatuurstudie

De literatuurstudie definieert eerst de gebruikte terminologie voor werkvloerbesturing en bakent daarmee het onderzoeksgebied voor deze thesis af. De rest van de literatuurstudie is geordend rond de tegenstelling tussen de hiërarchische en heterarchische besturing.

### 2.1 Werkvloerbesturing: fijnplanning en on line besturing

Het onderzoeksgebied van deze thesis is beperkt tot de besturingslogica voor werkvloerbesturing. Werkvloerbesturing (*shop floor control*) is het korte-termijngedeelte van het bedrijfsbeheer dat zich bezig houdt met de productie. Het omvat het beslissingsproces om beschikbare productiemiddelen toe te kennen aan de uit te voeren operaties, in een geschikte volgorde. Een van de belangrijkste taken van werkvloerbesturing is het nemen van de besturingsbeslissingen, kortweg de ‘besturingslogica’ (*manufacturing control*) of ‘productiemiddelentoekenning’ (*resource allocation*) genoemd.

Een typisch werkvloerbesturingssysteem omvat een fijnplanner en een on line besturing. Beide subsystemen nemen soortgelijke beslissingen (toekenning van machines en een starttijd aan alle uit te voeren operaties), maar de fijnplanner probeert de planning zoveel mogelijk te optimaliseren. De on line besturing (ook wel *dispatcher* genoemd) moet de planning uitvoeren en daarbij alle beslissingen *in real time* nemen.

Een belangrijk aspect van fijnplannen is dat dit dikwijls een NP-compleet probleem is. Dit betekent dat de rekentijd nodig om tot een optimale oplossing te komen zeer sterk stijgt (exponentieel of sneller) naarmate de parameters stijgen die de grootte van het probleem bepalen. Voor reële problemen betekent NP-compleetheid meestal dat het niet haalbaar is om een optimale oplossing te zoeken. Het heeft echter wel zin om te zoeken naar verbeteringen van de bestaande oplossing, ook al moet men niet hopen de optimale oplossing te vinden. Omdat dit optimalisatieproces toch wel rekenintensief is, wordt fijnplannen meestal op voorhand (off line) uitgevoerd.

### 2.2 Hiërarchische besturing

De traditionele aanpak voor CIM (*computer integrated manufacturing* — computergeïntegreerde productie) is gebaseerd op het principe van hiërarchische besturing. In een dergelijk systeem worden de uit te voeren taken opgesplitst in deeltaken, en elke

deeltaak weer in kleinere deeltaken. Na een functionele analyse komt men tot een aantal modules, die gestructureerd zijn in een boomvormige hiërarchie. Tussen de modules wordt enkel gecommuniceerd in verticale richting: een zogenaamde meestermodule geeft bevelen aan zijn slaafmodules, die de bevelen strikt opvolgen en de resultaten en sensorinformatie terugkoppelen naar de hogere niveaus.

De onderzoeksresultaten in verband met hiërarchische besturing refereren meestal naar de ontwikkeling van referentiearchitecturen voor werkvloerbesturing. Zo ontwikkelde NIST de AMRF-architectuur (Jones, 1986, Senehi, 1994), en het COSIMA-project de PAC-architectuur (Bauer, 1991). Met deze architecturen werden behoorlijke resultaten behaald, maar het bleek dat een dergelijke aanpak leidt tot strakke en rigide software, die moeilijk aanpasbaar is. Ook op operationeel niveau zijn er problemen met hiërarchische besturing, omdat ze traag reageren op storingen en het productiesysteem op die manier verlamd wordt tot de storing is opgelost.

Typische bouwblokken in een hiërarchische structuur zijn een fijnplanner en een dispatcher, die de planning uitvoert door de juiste machines op het gepaste moment aan te sturen. Op die manier ontstaat reeds een drielaagige hiërarchie, met een commandostroom die naar beneden gaat en een informatiestroom die naar boven gaat.

Fijnplanningsalgoritmen vormen al tientallen jaren een van de onderwerpen van operationeel onderzoek. In de eerste instantie zoekt men naar snellere algoritmen of algoritmen die een hogere performantie leveren. De bereikte resultaten gaan van snelle maar duidelijk sub-optimale dispatchingregels over kennisgebaseerde methoden (zoals ISIS, OPIS en CORTES, ontwikkeld aan Carnegie Mellon University), omgevingszoekalgoritmen, metaheuristieken (zoals *simulated annealing* en genetische algoritmen) tot theoretisch behoorlijk onderbouwde technieken zoals Lagrange-relaxatie. Een algemeen principe is wel dat de kwaliteit van de resulterende planning toeneemt met de rekentijd die hoort bij een bepaald algoritme.

Sinds duidelijk werd dat de resultaten van planningsonderzoek nauwelijks gebruikt werden, onder andere vanwege de problemen met storingen, is de focus van het onderzoek uitgebreid naar onderwerpen als reactieve fijnplanning. (Reactieve fijnplanning houdt in dat men een bestaande planning aanpast aan veranderende omstandigheden.) Ook op architectureel gebied is men de problemen met strikte hiërarchische systemen beginnen aanpakken door de ontwikkeling van gemodificeerd-hiërarchische architecturen. Dergelijke systemen laten ook communicatie toe tussen modules die niet in een meester-slaafrelatie tot elkaar staan en geven de lagere modules een zekere vorm van autonomie. Ze blijven echter vasthangen aan een duidelijke taakverdeling tussen alle modules, hetgeen de problemen van rigiditeit niet fundamenteel oplost.

Het aspect on line besturing is in veel mindere mate bestudeerd dan fijnplanning. In een hiërarchisch systeem is de taak van zo een besturing dan ook duidelijk afgebakend en vergt het ook niet zoveel inspanning om die besturingen te implementeren. Het meeste onderzoek spitste zich dan ook meestal toe op de softwareaspecten van de besturingen of het gebruik van formele modellen voor discretegebeurtenissystemen, zoals Petrinetten.

### 2.3 Heterarchische besturing

Toen duidelijk werd dat hiërarchische systemen moeite hadden om te reageren op storingen, ontstond een compleet nieuwe benadering voor de sturing van productiesystemen: heterarchische besturing. Heterarchische besturing is een in hoge mate gedistribueerde besturing die elke vorm van hiërarchie afzweert. Zo een besturing wordt geïmplementeerd door een groep autonome agenten — aparte, samenwerkende computerprocessen met elk hun eigen initiatief — met een hoge mate van autonomie, i.e. zonder gecentraliseerde controle.

De eerste publicaties over heterarchische besturing beschreven vooral de concepten, of simpele implementaties op basis van dispatchingregels (Vamos, 1983, Hatvany, 1985, Duffie, 1986). Zij wezen op de eenvoud van implementatie en het robuust gedrag onder storingen. Ook werd de analogie tussen multi-agentsystemen en marktprincipes uitgewerkt. Het *contract net protocol* definieert een coördinatieprotocol om het werk te verdelen onder de agenten. Vele algoritmen baseren zich ook op de wet van vraag en aanbod, om zo capaciteitsbeperkingen te modelleren en op te leggen (uitgetest door Debels et al., 1998). Een aantal technieken gebruiken marktmechanismen ook voor optimalisatie (Luh, 1993, Gou, 1994), hetgeen vaak dichter aanleunt bij gedistribueerde fijnplanning.

Eén van de meest veelbelovende aspecten van gedistribueerde systemen is het potentieel van emergent gedrag: door de interactie van eenvoudige gedragingen (in de verschillende agenten) ontstaat een niet-triviale (maar gewenste) nieuwe eigenschap van het systeem. Een voorbeeld daarvan is de zogenaamde ‘onzichtbare hand’, bekend in de economie (en waarop de marktmechanismen voor productiesturing gebaseerd zijn): door het invoeren van machineprijzen moet er aan de capaciteitsbeperkingen niet expliciet voldaan worden, maar wordt vraag en aanbod in evenwicht gehouden door de agenten (leveranciers en klanten) in de markt. Een ander voorbeeld van mogelijk emergent gedrag is gebaseerd op het gedrag van mieren. Veelbelovend is ook het werk op bionische productiesystemen, die potentiaalvelden en genetische algoritmen gebruiken voor de sturing van de werkvloer. Meer toegepast is het werk van Tharumarajah (1997), die expliciet fijnplanningsproblemen aanpakt.

Toch zijn niet alle problemen opgelost met heterarchische besturing. Het is niet gemakkelijk om dit concept toe te passen op echt grote problemen (bijvoorbeeld het beheer van een hele fabriek, of een hele werkvloer), dikwijls vanwege bijvoorbeeld de communicatie-overhead. Een fundamenteeler probleem is echter de inherente onvoorspelbaarheid van heterarchische systemen, en het feit dat ze meestal geen mogelijkheid toestaan tot globale optimalisatie.

### 2.4 Conclusie

Zowel hiërarchische als heterarchische systemen vormen waardevolle bijdragen tot de huidige stand van het onderzoek. Om echter optimalisatie en robuustheid tegen storingen te combineren, moet er gezocht worden naar een combinatie van beide benaderingen. Dat is juist het doel van deze thesis.

### 3. Holonische productiesystemen

Holonische productiesystemen (*Holonic Manufacturing Systems* — HMS) zijn sterk gedecentraliseerde productiesystemen die opgebouwd worden uit een verzameling modulaire, gestandaardiseerde, autonome, samenwerkende en intelligente bouwstenen, "holons" genaamd.

Het HMS concept is gebaseerd op het pionierswerk van Arthur Koestler (1967) en Nobelprijswinnaar Herbert Simon over complexe systemen. Hierbij introduceerde Koestler het woord "holon", dat een samenvoeging is van het Griekse woord "holos" (het geheel) en het achtervoegsel "-on" wat op een deeltje wijst zoals in proton en neutron. De toepassing van de concepten van holonische systemen op productie werd uitgewerkt binnen het IMS-programma, een wereldwijd onderzoeksprogramma voor productiesystemen.

Twee essentiële aspecten van dergelijke holons zijn hun autonomie en hun vermogen tot samenwerken. Hierbij dient de autonomie tot het lokaal verwerken van storingen zonder de ondersteuning door hiërarchisch hoger gelegen holons. Anderzijds zorgen samenwerkingsmechanismen ervoor dat een globale optimalisatie door het systeem wordt nagestreefd. In de loop van het doctoraatsonderzoek bleek dat mijn ontwikkelingen — binnen het beperkt kader van productiesturing voor flexibele productiesystemen — parallel liep met het HMS-project. In die zin zijn vele resultaten van mijn thesis concrete uitwerkingen van de soms abstracte ideeën waarmee het HMS project gestart was.

Aan de K.U.Leuven (afdeling PMA) werd de voorbije jaren heel wat onderzoek verricht naar holonische productiesystemen. Wyns (1999) ontwikkelde een referentiearchitectuur voor holonische productiesystemen. Die bestaat uit een basisstructuur van product-holons, orders-holons en productiemiddel-holons, die gestructureerd worden aan de hand van object-georiënteerde concepten zoals specialisatie en aggregatie. Die basisstructuur wordt aangevuld met stafholons, gecentraliseerde experten die de holons in de basisstructuur advies geven in verband met gespecialiseerde taken of overkoepelende problemen. Ikzelf heb onderzoek verricht naar productiesturing. Vanginderachter en Tanaya (Kruth, 1996) werken aan een holonische on line werkvoorbereider en NC-controller.

Naast autonomie en samenwerking is in die samenwerkingsverbanden ook gebleken dat de mogelijkheid tot evolutionaire ontwikkeling noodzakelijk is in een holonisch systeem. Bij het ontwerp van holonische modules mag men zich niet op de architectuur baseren. Om een vlotte herconfigureerbaarheid te bekomen moet de realisatie van de architectuur de laatste fase zijn van het systeemontwerp en niet het uitgangspunt. Door hiërarchie toe te laten in een systeem van autonome elementen, zijn complexe, adaptieve systemen gemakkelijker te ontwerpen.

Praktisch gezien worden holonische productiesystemen geïmplementeerd met multi-agenttechnologie, m. a. w., holons zijn agenten die werken volgens de holonische concepten. Ook objectoriëntatie, softwareherbruikbaarheid en herconfigureerbaarheid zijn belangrijke aspecten. Om dat te bereiken wordt ook gedacht aan meer geavanceerde technieken zoals machine-leren en zelf-organiserende systemen. Maar

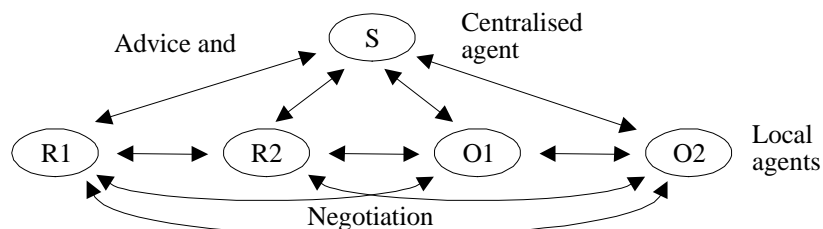
ook de mens, als meest flexibele en intelligente factor in het productieproces, maakt inherent deel uit van het holonisch productiesysteem.

## 4. Generisch model voor holonische productiebesturing

Dit hoofdstuk beschrijft het generisch gedeelte van de resultaten van deze thesis. Het begint met een aantal algemeen bruikbare concepten voor holonische werkvloerbesturing. Daarna wordt een statisch model (een besturingsarchitectuur) voorgesteld, dat de structuur van het besturingssysteem beschrijft. Een volgende paragraaf toont hoe de verdeling van beslissingsbevoegdheid kan beschreven worden. Tenslotte toont een dynamisch model hoe de verschillende holons samenwerken. Samen vormen die concepten en modellen het generisch model voor holonische productiebesturing.

### 4.1 Algemene concepten

Om robuustheid tegen storingen te combineren met de voorspelbaarheid en optimalisatie van performantie, wordt voorgesteld om hiërarchische elementen te integreren in een gedistribueerd (multi-agentgebaseerd) besturingssysteem (figuur 1). De resulterende besturingsarchitectuur heeft dan een aantal autonome lokale holons, die door onderlinge onderhandeling het werk onder elkaar wel kunnen verdelen, maar daarin bijgestaan worden door gecentraliseerde holons, stafholons genaamd.



Figuur 1. Hierarchie in gedistribueerde systemen. R1 en R2 zijn machineholons (resource holons), O1 en O2 zijn orderholons. S is een stafholon.

Om dit concept te motiveren, is het interessant om te kijken naar de mechanismen die hiërarchische en heterarchische systemen hun goede eigenschappen geven. Het emergente en reactieve gedrag van heterarchische systemen ontstaat juist door de interacties tussen de individuele gedragingen van elke agent / elk holon. Hiërarchische systemen halen hun voorspelbaarheid juist door de coördinerende rol die

het centrale element speelt, en hun hoge performantie doordat dat centraal element tijd en moeite steekt in een optimalisatieproces. Bovendien hebben zowel lokale als gecentraliseerde holons hun eigen vaardigheden en opportuniteiten. Lokale holons hebben meer up-to-date en gedetailleerde informatie over de entiteiten die ze beheeren, en kunnen daarom ook sneller en gepaster reageren op storingen. Maar gecentraliseerde holons hebben een overzicht over het hele systeem, wat hun toelaat om gecoördineerd acties uit te voeren en de implicaties op de globale performantie te doorgronden. Het loont dus de moeite om te onderzoeken of de combinatie van hiërarchische en heterarchische systemen ook de voordelen van beide oplevert.

Een tweede concept van deze thesis is het verdelen van beslissingsbevoegdheid over alle holons, en dat op verschillende niveaus. Dit komt eigenlijk neer op de ondubbelzinnige toepassing van het vorige concept. Om hiërarchie in gedistribueerde systemen in te voeren zonder de beslissingsbevoegdheid effectief te distribueren, zou men afwisselend hiërarchisch en heterarchisch moeten werken. Maar het resulterende systeemgedrag zou eerder houderig zijn, en zou ook de nadelen van hiërarchische en heterarchische systemen combineren, zelfs als men op het juiste moment zou switchen tussen beide werkingsmodes. Door beslissingsbevoegdheid te verdelen, ontstaat er een zachte overgang tussen hiërarchische en heterarchisch gedrag, waarbij alle holons bijdragen tot de beslissingen volgens hun eigen kennis en vermogen. Belangrijk is dat de holons zo kunnen samenwerken zonder in een meester-slaafrelatie te vervallen.

Tenslotte wordt ook het concept voorgesteld om gelijktijdig te fijnplannen en die planning uit te voeren. Zo kan snel gereageerd worden op storingen terwijl de fijnplanner toch de tijd gegeven wordt om zijn planning te optimaliseren. Bemerkt dat dit concept tot nog toe weinig toegepast werd: ofwel werd het planningsalgoritme sterk vereenvoudigd (en versneld) zodat de reactie op storingen snel genoeg was (maar ten koste van de performantie); ofwel was de on-line besturing zeer eenvoudig, waardoor de reactie op storingen ver van optimaal was. Bij het gelijktijdig optimaliseren en uitvoeren van een planning wordt er op verschillende niveaus gereageerd op storingen. De lokale holons reageren onmiddellijk, zodat de productie niet stilvalt, maar ook de planner zal reageren op de storing en zijn planning eraan aanpassen, ook al is dat niet ogenblikkelijk.

Dit concept kan wel niet zonder verder onderzoek toegepast worden. Het vergt immers de samenwerking tussen een reactieve fijnplanner en een intelligente on-line-werkvloerbesturing. Reactieve-planningsalgoritmen vormen geen probleem, want zij worden al enkele jaren bestudeerd, en er bestaat dan ook genoeg literatuur over, met inbegrip van heel wat algoritmen en numerieke resultaten. Er zijn echter weinig of geen algoritmen voor een intelligente on-line-werkvloerbesturing. Er is nood aan algoritmen die *een fijnplanning autonoom kunnen uitvoeren*, wat betekent dat ze die planning kunnen volgen onder normale condities, maar ervan afwijken (en op de juiste manier) wanneer nodig. Deze thesis levert enkele bijdragen op dit gebied, maar er is nog heel wat werk te doen. Tenslotte is ook de samenwerking zelf belangrijk en niet altijd triviaal, omdat beide systemen verder evolueren terwijl ze de



planning optimaliseren en uitvoeren, wat betekent dat ze een model van elkaars gedrag nodig hebben.

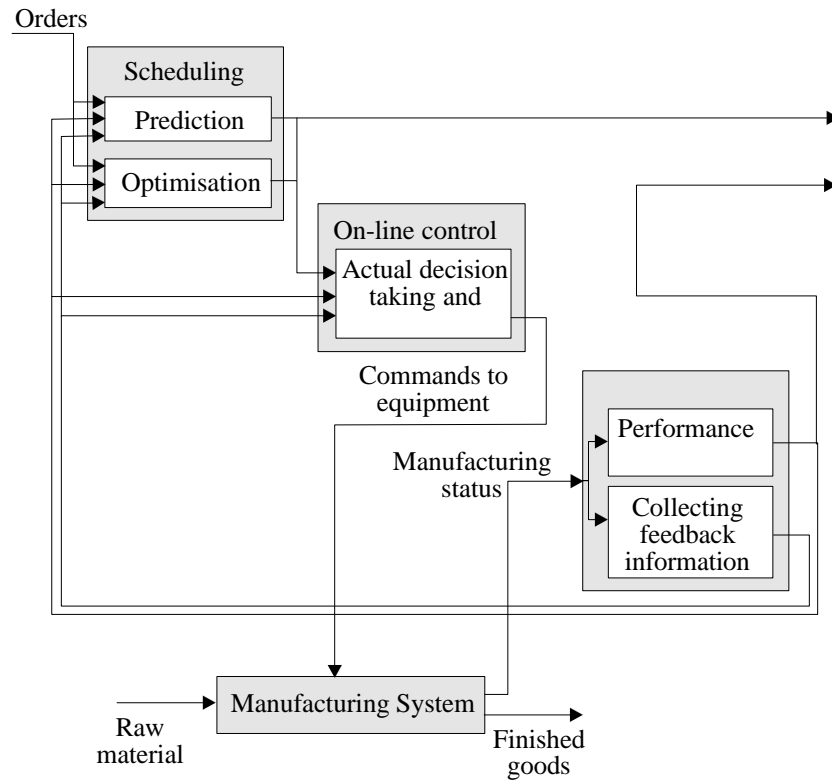
## 4.2 Statisch model

Om de architectuur van een holonisch besturingssysteem te beschrijven is een statisch model nodig van de holons in het systeem. Dit model identificeert welke functies nodig zijn om een holonisch productiesysteem te besturen en met welke holons die functies uitgevoerd worden. Zo een model schetst ook de relaties tussen de holons en de manier waarop ze zullen samenwerken (maar voor een volledig en accuraat beeld van de architectuur moet ook de dynamica beschreven worden — zie verder).

De functies die nodig zijn om een werkvloer te besturen zijn planning, sturing en monitoring (figuur 2). Primordiaal is natuurlijk de sturing: het nemen van de beslissingen om operaties op een welbepaalde machine in een welbepaalde volgorde uit te voeren en het starten van machines en transportmiddelen om die beslissingen te realiseren. Planning houdt twee aspecten in: enerzijds is het bij voorkeur op voorhand geweten hoe het systeem zich zal gedragen; anderzijds moet ook de prestatie geoptimaliseerd worden. Monitoring is essentieel om te reageren op storingen en de planning aan te passen aan de nieuwe omstandigheden.

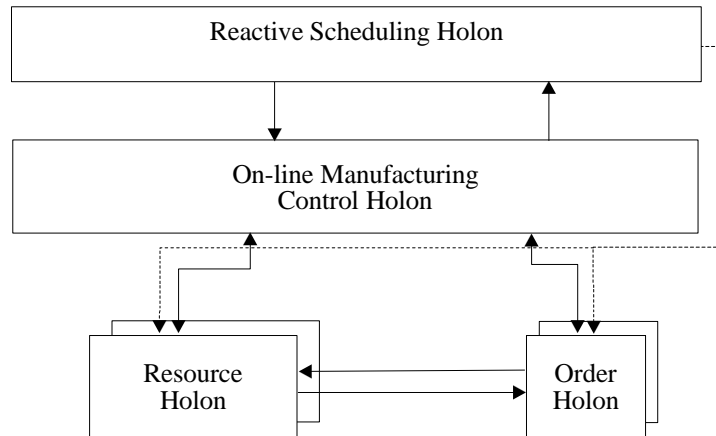
Al die functies worden geïmplementeerd in een groep van werkstation-, transportsysteem- en orderholons, geassisteerd door een on-line-besturingsholon en een fijnplanningsholon (figuur 3). De basisfunctionaliteit moet gegarandeerd worden door de lokale holons, i.e. de productiemiddel- en de orderholons. Door middel van gekende onderhandelingsprotocollen (bvb. het *contract net protocol*) moeten zij er steeds voor kunnen zorgen dat de productie voortgaat. Het on-line- besturingsholon suggereert echter al aan de lokale holons welke beslissingen zij best nemen, rekening houdend met een globaal overzicht over de werkvloer. In die zin handelt het net zoals een ploegbaas het werk van zijn arbeiders coördineert. OP zijn beurt wordt het on-line-besturingsholon weer bijgestaan door het advies van een fijnplanningsholon, die de toekenning van productiemiddelen aan operaties optimaliseert.

Belangrijk is inderdaad dat het nemen van een beslissing niet overgelaten wordt aan één holon, maar dat ze —in overeenstemming met concept 2 — door alle holons samen genomen wordt, waarbij elk holon zijn eigen bekwaamheden gebruikt. Ook het optimaliseren van de planning wordt niet volledig overgelaten aan de fijnplanner: snelle verbeteringen kunnen ook uitgevoerd worden door het on-line-besturingsholon, en lokale optimalisaties kunnen perfect door een werkstation of order uitgevoerd worden. Anderzijds moet ook de fijnplanner reageren op storingen en zich aanpassen aan de nieuwe omstandigheden, zij het dan niet zo snel als de lokale holons.



Figuur 2. Functies in een werkvloerbesturing.

Een belangrijk kenmerk van een holonische besturing is dat de structuur van het systeem gemakkelijk, zelfs tijdens productie, moet kunnen veranderen. (Het is niet omdat het model van figuur 3 statisch is, dat de structuur zelf niet kan veranderen.) Dit laat toe om de meest adequate structuur voor een productiesysteem kan opgelegd worden al naargelang de omstandigheden. In diezelfde optiek is het mogelijk om verschillende fijnplanners te combineren, en zelfs om gedistribueerde fijnplanners te introduceren.



Figuur 3. Statisch model van de holons in een holonische productiebesturing.

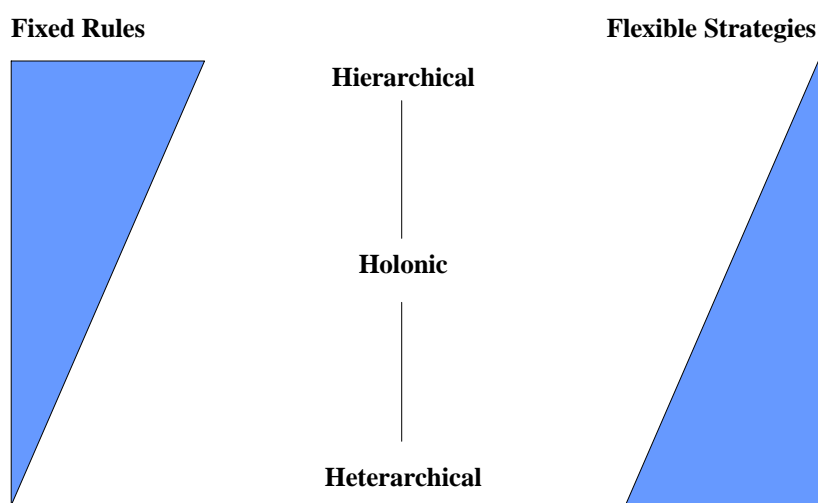
### 4.3 Vaste regels en flexibele strategieën.

In holonische productiesystemen, waarin elke uit te voeren functie niet toegekend wordt aan één holon, is het noodzakelijk om de verdeling van de beslissingsbevoegdheid over de verschillende holons nauwkeurig te modelleren. Inspiratie voor zo een model werd gevonden bij het werk van Koestler (1967): toen hij het begrip *holon* definieerde, beschreef hij ook hoe de interacties tussen individuen in een complexe samenleving georganiseerd worden. Sociale interacties worden voor een groot gedeelte bepaald door vaste afspraken, regels en gewoonten, en zijn soms maar voor een klein gedeelte het gevolg van de individuele wil. Dit beperkt de autonomie van het individu, maar zorgt er anderzijds ook voor dat de groep een zekere mate van stabiliteit en cohesie heeft. Bovendien blijven er nog heel wat keuzen over, die elk individu op zijn eigen manier “volgens flexibele strategieën” kan invullen. Het is boeiend om vast te stellen dat de mate waarin vaste regels of flexibele strategieën het gedrag van een individu bepalen, sterk afhangt van het type van organisatie waarin die persoon zich beweegt: in een hiërarchische organisatie is het aandeel van de vaste regels behoorlijk groot, terwijl in lossere groepen het belangrijk is om flexibel zijn eigen strategie te ontwikkelen.

Deze analogie met menselijke organisatie toont dadelijk hoe de verdeling van beslissingbevoegdheid kan gemodelleerd worden. Om de samenwerking tussen en mate van autonomie van de holons te definiëren moet men vastleggen welke vaste regels de holons moeten volgen. De resterende vrijheid kan dan ingevuld worden met de flexibele strategieën van elk holon. Dit mechanisme laat toe om geleidelijk en

op een vlotte manier over te gaan van strikt hiërarchische naar volledig heterarchisch besturing (figuur 4).

Het kiezen van vaste regels heeft een soortgelijk effect als het definiëren van invarianten bij het schrijven van algoritmen: het vormt een hoogniveau-ontwerp van een holonische samenwerkingsmechanisme. Het definiëren van de vaste regels (vanaf nu ook wel invarianten genoemd) legt het ontwerp van een holonische besturingsstrategie voor een groot deel vast.



Figuur 4. Evenwicht tussen invarianten (Vaste regels) en Varianten (Flexibele Strategieën) onder verschillende soorten besturing.

#### 4.4 Dynamisch model

Om autonoom te reageren op storingen en samen te werken met andere holons, moet een holon verschillende taken in parallel uitvoeren. Die holons werken asynchroon met de andere holons, maar hebben toch heel wat synchronisatiepunten, waar ze informatie uitwisselen, onderhandelen, advies geven, e.d.

Om een goed samenwerkingschema te ontwerpen, is in de eerste plaats een model nodig om zo een ontwerp te beschrijven. In deze thesis is geopteerd voor het gebruik van Petrinetten. Zij modelleren dit soort discrete-gebeurtenissystemen goed en kunnen ook het tijdsaspect vatten (met behulp van tijds-gerelateerde Petrinetten). Eenmaal de samenwerking gemodelleerd, is het trouwens ook eenvoudiger om de parameters ervan te regelen en te optimaliseren.

Een voorbeeld van een dergelijk samenwerkingsmodel is gegeven in het volgende hoofdstuk. Het handelt over de modellering van gelijktijdige fijnplanning en planningsuitvoering.

## 5. Samenwerking in holonische productiebesturing

Dit hoofdstuk beschrijft aan de hand van enkele voorbeelden hoe samenwerking in een holonische productiesturing kan ontworpen worden. Deze voorbeelden zijn niet generisch voor alle holonische oplossingen, maar geven wel een concreet beeld van de besturing. In deze samenvatting wordt de meeste aandacht gegeven aan de samenwerking tussen het fijnplanningsholon en het on-line-besturingsholon, die gelijktijdig een fijnplanning berekenen respectievelijk uitvoeren. Andere voorbeelden worden — zeker in de samenvatting — minder gedetailleerd overlopen.

Voor het modelleren van de samenwerking wordt gebruik gemaakt van Petrinetten, die de uit te voeren taken van elk holon op een hoog niveau van aggregatie beschrijven. Naargelang de noodzaak worden de Petrinetten uitgebreid met tijd, voorwaarden, stochasticiteit, besturingsplaatsen, e.d.

### 5.1 Gelijktijdig berekenen en uitvoeren van een fijnplanning

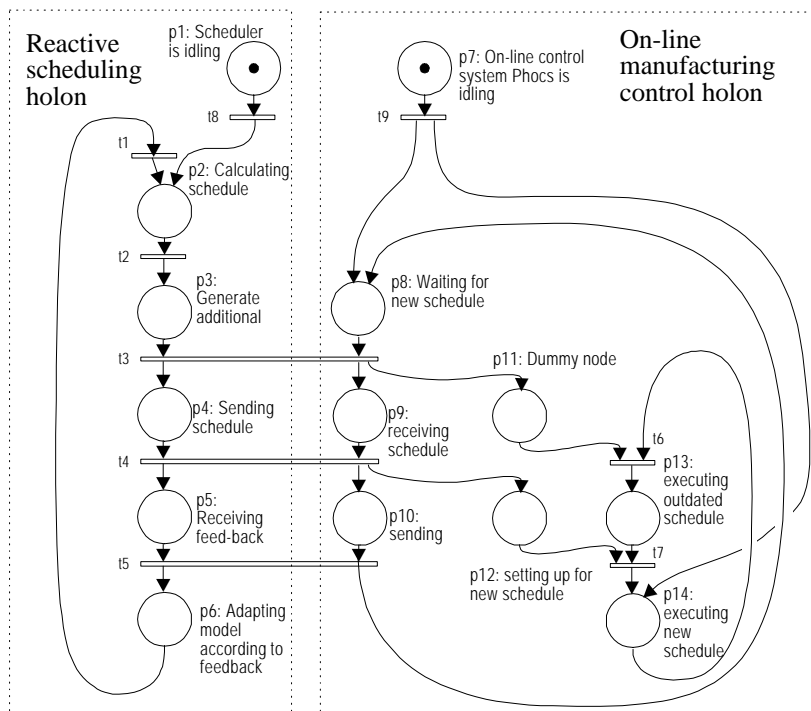
Deze paragraaf beschrijft het concept van gelijktijdige fijnplanning en planningsuitvoering met behulp van een tijdsgerelateerd Petrinet (figuur 5). Zo een model beschrijft de samenwerking tussen het fijnplanningsholon PaRSiFAL en het on-line-besturingsholon PHOCS, zoals die ontwikkeld werden voor deze thesis (zie hoofdstuk 7). De fijnplanner en het on-line-besturingsholon werken continu, gelijktijdig en in parallel, en asynchroon. De fijnplanner gebruikt dus alle beschikbare tijd om de planning te optimaliseren. Op regelmatige tijdstippen contacteert hij het on-line-besturingsholon om een update van de planning door te geven en zelf *feedback* van het productiesysteem te krijgen. De fijnplanner past zijn planning daaraan aan en herbegint de optimalisatie (want de vorige planning is onder gewijzigde omstandigheden natuurlijk niet meer optimaal). Ondertussen bereidt het on-line-besturingsholon zich voor om de nieuwe planning uit te voeren. Het kan dat niet onmiddellijk, want de planning omvat operaties die een voorafgaandelijke transport- of insteloperatie nodig hebben. (Dit betekent ook dat het on-line-besturingsholon twee versies van de planning nodig heeft, die het naadloos in elkaar moet laten overgaan.)

Zo een Petrinet beschrijft ondubbelzinnig wat het concept van gelijktijdige planning en planningsuitvoering eigenlijk inhoudt. Zijn structuur toont ook duidelijk hoe dit concept geïmplementeerd kan worden. Vooral PHOCS moet in zijn implementatie verschillende *threads* hebben.

Tijd is in figuur 5 gemodelleerd door aan elke plaats van het Petrinet een minimum verblijftijd toe te kennen, voor de daaropvolgende transitie kan vuren. Dit is een logische manier om 'tijd' te modelleren in dit Petrinet, vermits de tijden dan overeenkomen met de duur van een hoog-niveautaak. Deze tijden zijn belangrijke parameters van het samenwerkingsprotocol. De belangrijkste parameter is de reken-tijd van de fijnplanner, en nauw verbonden daarmee, de communicatiefrequentie. Het bepalen van de optimale waarde daarvan is een trade-off tussen de mate van op-

## Samenvatting

timaliteit van de (her-)planning en de reactiesnelheid op storingen. De tijd nodig voor communicatie (zenden en ontvangen) kan beschouwd worden als een constante parameter (gegeven de probleemgrootte en de snelheid van het intranet). De tijd die PHOCS nodig heeft om zich aan te passen aan de nieuwe planning is in principe ook een gegeven constante. Door deze tijd optimistisch in te schatten kan de globale systeemperformantie echter verbeteren, zodat deze tijd een regelbare parameter wordt. In de huidige implementatie is die waarde gekozen op 80 % van de nodige waarde. Interessant is ook dat de analyse van al deze tijden bevestigt dat het systeem (en bijhorend Petrinet) in de meeste omstandigheden periodisch en begrensd is.



Figuur 5. Tijdsgerelateerd Petrinetmodel van de samenwerking tussen het on-line-besturingsholon PHOCS en de reactieve fijnplanner PaRSiFAL.

## 5.2 Samenwerking tussen lokale holons en stafholons

Een ander samenwerkingsmechanisme dat bestudeerd werd in de thesis is dat tussen lokale holons en stafholons. Lokale holons zijn in principe in staat om door onderhandeling zelf tot een oplossing te komen, maar kunnen daarin bijgestaan worden door stafholons. Dit vergt een extensie van het contractnetprotocol, dat voorziet in een onderhandelingsmechanisme tussen aanbieders en bidders. In een holonisch systeem is er ook een plaats voor een bemiddelaar, zoals het stafholon. Een model daarvoor wordt besproken in de volledige tekst.

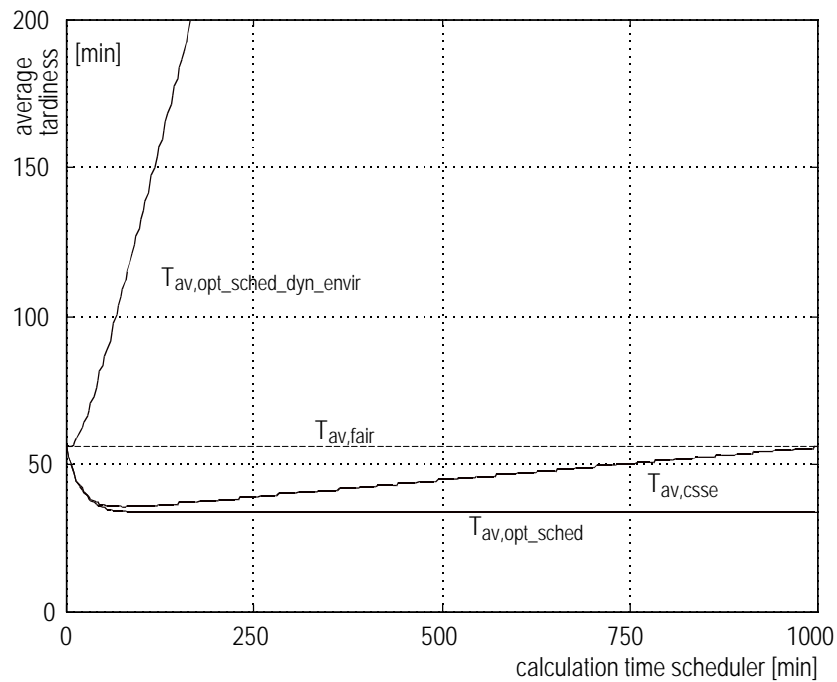
## 5.3 Herplannen na belangrijke gebeurtenissen

Een verbetering van het voorgaande samenwerkingsprotocol tussen fijnplanner en on-line-besturingsholon bestaat erin om niet alleen periodisch, maar ook na belangrijke gebeurtenissen te herplannen. Het idee is dat het bij belangrijke storingen, zoals nieuwe orders of machineuitval, niet interessant is om te blijven verder rekenen aan de planning die toch al verouderd is. Dit houdt in dat er moet geïdentificeerd worden wat een belangrijke storing is, en hoelang er dan opnieuw moet geoptimaliseerd worden. De thesis presenteert een ontwerp van een dergelijk samenwerkingsmechanisme, echter zonder het in detail uit te werken.

## 5.4 Optimalisatie van parameters

Een goede keuze voor de parameters van de samenwerking tussen holons is vaak belangrijk voor de uiteindelijke performantie van het productiesysteem. In deze thesis werd, als voorbeeld, uitgewerkt hoe de keuze van de rekentijd kan geoptimaliseerd worden om de samenwerking tussen reactieve fijnplanner en on-line-besturingsholon zo goed mogelijk te regelen.

Het is moeilijk om die optimalisatie analytisch uit te voeren voor het complete systeem. In plaats daarvan wordt voor die optimalisatie uitgegaan van een vereenvoudigd model. Met een aantal veronderstellingen kan men de gemiddelde laattijdigheid (voor een niet-geoptimaliseerd systeem) berekenen. Bij optimalisatie zal de performantie van de planning verbeteren (dus de laattijdigheid afnemen) in functie van de rekentijd. Houdt men er dan rekening mee dat de planning later zal uitgevoerd worden als de rekentijd toeneemt, dan ziet men dat uiteindelijke performantie van het systeem eerst verbetert bij toenemende rekentijd, maar later weer verslechtert, omdat de extra verbeteringen door langer te rekenen niet opwegen tegen de toenemende laattijdigheid door de planning later te starten. Met een eenvoudig model om de performantie in te schatten van gelijktijdige planning en planningsuitvoering kan men vergelijken wat de performantie is van de vier systemen, zoals getoond in figuur 6. De resultaten onder de meest optimistische omstandigheden zijn veelbelovend, namelijk dat sequentiële planning en planningsuitvoering geen voordeel levert ten opzichte van niet-optimalisatie, en bij gelijktijdige planning en planningsuitvoering de performantie met 90 % van de optimale mogelijke waarde verbeterd wordt. Onder zeer voorzichtige omstandigheden zakt het verschil tussen sequentiële en gelijktijdige planning en planningsuitvoering tot slechts 4 %, wat klein, maar toch nog significant is. Experimenten zullen aantonen (zie hoofdstuk 8) dat de waarheid ergens in het midden ligt.



Figuur 6. Kwantitatieve vergelijking van off-line-planningsmethoden, dispatching, sequentieel plannen en planningsuitvoeren en van gelijktijdig plannen en planningsuitvoeren. Dit voorbeeld is gebaseerd op de meest optimistische formulering van het probleem, maar geeft goed de mogelijke verbeteringen aan. Voor verdere gegevens wordt verwezen naar de volledige beschrijving in hoofdstuk 5.

## 6. Holonische besturingsalgoritmes

Bij het zoeken naar holonische besturingsalgoritmes is de aandacht vooral gegaan naar planningsuitvoering, vermits er al heel wat planningsalgoritmes bestaan die reageren op storingen. Dit hoofdstuk beschrijft een aantal van die algoritmes op basis van het kader van vaste regels en flexibele strategieën, zoals dat gedefinieerd is in hoofdstuk 4. Een ander algoritme, op basis van perturbatie-analyse, bouwt zijn ge-



drag op een theoretisch beter uitgewerkt kader, en bedient zich niet expliciet van die regels en strategieën.

### **6.1 Definitie planningsuitvoering**

Planningsuitvoering is de functie die inhoudt dat de toekenningen van operaties aan machines en de volgordes van operaties zoals die voorgesteld worden door de planning, ook effectief worden uitgevoerd, maar op een intelligente manier: als er storingen zijn, mag van de planning afgeweken worden.

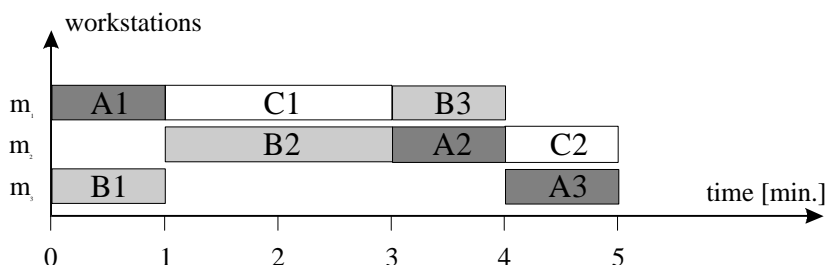
Een on-line-besturingsalgoritme doet aan autonome planningsuitvoering als het door gebruik te maken van de planning, duidelijk sneller kan reageren op storingen dan een reactieve planning, en toch de optimalisatie van de planning kan gebruiken. In andere woorden, de uiteindelijke performantie moet gecorreleerd zijn aan de performantie van de fijnplanner.

Het is belangrijk te vermelden dat autonome planningsuitvoering geen eenvoudige taak is. Eigenlijk is het onmogelijk om op een optimale manier een planning uit te voeren (als er storingen optreden, en zonder zelf te herplannen), als de planning enkel voorgesteld wordt als een Ganttkaart. Dat kan aangetoond worden met een voorbeeld. In dat voorbeeld wordt een storing geïnduceerd bij de uitvoering van een planning, waarbij het niet meteen duidelijk is wat de juiste reactie op die storing is. Het blijkt dat de juiste keuze afhangt van de relatieve dringendheid van verschillende operaties. Maar die dringendheid hangt zelf weer af, gekoppeld via een netwerk van andere operaties, van de uiteindelijke laattijdigheid die orders eventueel zouden oplopen ten gevolge van die storing. In deze thesis wordt een oplossing gezocht voor dit probleem door het invoeren van extra advies van de fijnplanner aan het on-line-besturingsholon.

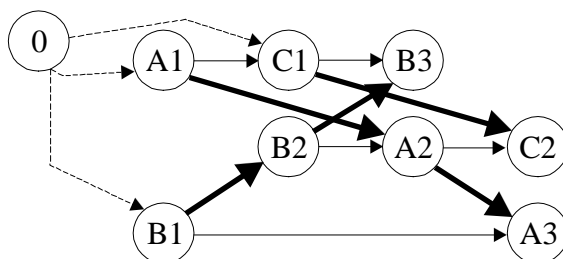
### **6.2 Planning voorgesteld door een grafe**

Zowel voor het definiëren van vaste regels en flexibele strategieën als voor het ontwerpen van autonome planningsuitvoeringsstrategieën kan het handig zijn om de planning voor te stellen als een grafe, en niet alleen door een Ganttkaart (figuur 7). Deze grafe bestaat uit knopen, die de operaties voorstellen, en bogen, die de volgordebeperingen tussen de operaties voorstellen (figuur 8). Die volgordebeperingen zijn natuurlijk zeker en vast de technologische volgordebeperingen, maar ook de beperkingen die volgen uit de volgorde van de operaties op een productiemiddel in de planning. Ook het bestaan van een vroegst toegelaten starttijd voor een order (*release date*) creëert een volgordebepering die in de grafe kan gemodelleerd worden.

Men kan aantonen dat de voorstelling van die planning als een grafe equivalent is aan de klassieke Gantt-kaart, als de originele planning een actieve planning is. Dit betekent dat de starttijd voor elke operatie onder beide voorstellingen dezelfde is. De starttijd van een operatie in een grafe kan berekend worden door de vroegst mogelijke starttijd van elke operatie te berekenen langsheen de pijlen van de grafe, te starten vanaf de vroegst toegelaten starttijd van elk order. Zo kan men ook de performantie berekenen van de hele planning.



Figuur 7. Een fijnplanning voorgesteld door een Gantt-kaart.



Figuur 8. Voorstelling van een fijnplanning als een grafe, in plaats van door een Ganttkaart, zoals in figuur 7.

### 6.3 Hiërarchische besturing

In een hiërarchische besturing volgen de onderste modules in de hiërarchie de bevelen van hun meester strikt op. Voor het uitvoeren van een planning betekent dit dat het on-line-besturingsholon de planning uitvoert zoals ze gegeven is. Als er storingen zijn, worden de volgordes opgevolgd zoals opgelegd door de planningsgrafe, ook als daarvoor machines en orders moeten wachten met de productie. Dit komt erop neer dat de samenwerking tussen fijnplanner en on line besturing volledig geregeld wordt door vaste regels, zonder dat er plaats blijft voor flexibele strategieën. Het on-line-besturingsholon moet in deze opstelling dus weinig intelligentie aan de dag leggen.

Hoofdstuk 8 toont experimenteel aan dat hiërarchische besturing een hoge performantie levert als er geen storingen zijn en als de planning goed geoptimali-

seerd is, maar dat deze performantie snel zakt als er storingen komen. Ook is het gedrag van hiërarchische besturing goed voorspelbaar.

#### **6.4 Heterarchische en gecentraliseerde autonome besturing**

In heterarchische systemen worden alle beslissingen genomen door autonome agenten, die met elkaar communiceren en onderhandelen om de productie te runnen. Daarbij zweert men elke vorm van hiërarchie af. Heterarchische besturing is veelbelovend wat betreft het reageren op storingen, omdat een reactief gedrag automatisch ontstaat uit de interacties tussen de verschillende agenten.

De onderzochte implementaties van heterarchische besturing zijn gebaseerd op vrije-marktmechanismen en klassieke dispatchingregels. Debels en Hermans (1998) implementeerden voor hun eindwerk op PMA een marktmechanisme dat de beslissingen van elk order deed afhangen van een machinekostprijs, een transportkostprijs en de levertermijn (*due date*). Zelf implementeerde ik een gecentraliseerde vorm van autonome besturing, waarbij orders machines kozen op basis van de laagste belasting, en machines een operatievolgorde kozen op basis van orders met de vroegste levertermijn (*earliest due date*). Hiervoor werden op een gecentraliseerde manier dezelfde beslissingsregels gebruikt als men normaal in een gedistribueerde besturing op basis van dispatchingregels zou gebruiken. Essentieel is natuurlijk dat in beide implementaties geen gebruik werd gemaakt van een voorspellende en optimaliserende fijnplanner. Het systeemgedrag werd dus volledig beschreven door de flexibele strategieën van de on-line besturing.

Het reactief gedrag tegen storingen wordt experimenteel bevestigd in hoofdstuk 8, maar daar wordt ook getoond dat heterarchische systemen een lagere performantie hebben en minder voorspelbaar zijn dan de klassieke hiërarchische systemen.

#### **6.5 Heuristische holonische planningsuitvoeringsalgoritmen**

Voor deze thesis zijn drie heuristische besturingsalgoritmen ontwikkeld die een fijnplanning autonoom kunnen uitvoeren doordat ze een gedeeltelijke autonomie krijgen. Ze worden voor een gedeelte gebonden door vaste regels, maar die laten hun nog voldoende vrijheid om zelf, volgens hun eigen flexibele strategieën, te reageren op storingen.

Een eerste algoritme (STA — *short term autonomy*) geeft het on-line-besturingsholon autonomie voor een korte termijn. Het mag alleen die operaties uitvoeren die gepland zijn in ‘de nabije toekomst’ (geparametriseerd door een parameter  $T_{resched}$ ), maar mag die operaties dan wel uitvoeren in een volgorde en op machines die het zelf kiest.

Een tweede algoritme (AID — *autonomy if disturbances*) krijgt autonomie als er storingen ontdekt worden. Dit wordt praktisch geïmplementeerd door beperkingen, opgelegd door de vaste regels, te laten wegvallen als ze verouderd zijn (eventueel getemperd of versneld door een positieve of negatieve buffertijd in te voeren die definieert wanneer een beperking verouderd is).

Een derde algoritme (LDFS — *limited deviation from schedule*) staat een beperkte afwijking van de planning toe. Dit betekent dat de on-line besturing de opera-

ties moet uitvoeren rond de geplande starttijd, maar dat omwisselingen van operaties kunnen zolang dat niet resulteert in grote afwijkingen van de planning.

In de volledige tekst worden die drie algoritmen formeel gedefinieerd door middel van de voorstelling van een planning als een grafe.

Deze drie algoritmen hebben met elkaar gemeen dat ze proberen de goede kenmerken van een planning te extraheren, zonder zich te houden aan minder belangrijke beperkingen als er storingen zijn. Op die manier zijn ze holonisch, doordat ze proberen robuustheid tegen storingen te combineren met het gebruik van een centraal geoptimaliseerde planning. Het blijven echter heuristische methoden, gebaseerd op gezond verstand. Hun uiteindelijke performantie moet dan blijken uit experimenten. Daarom beschrijft de volgende paragraaf ook een planningsuitvoeringsalgoritme op basis van een meer theoretisch gefundeerde afleiding.

### **6.6 Planningsuitvoering op basis van perturbatie-analyse**

Om een planning naar behoren uit te voeren, wil dit algoritme daadwerkelijk gebruik maken van extra advies van de planner. Er was reeds vermeld dat zo een advies noodzakelijk is om steeds tot goede beslissingen te komen, omdat er in een Ganttkaart essentiële informatie ontbreekt. Het probleem is dat een on-line besturing moet weten hoe lokale beslissingen de globale performantie beïnvloeden.

Dit algoritme gebruikt daarom perturbatie-analyse (PA). De fijnplanner stelt het resultaat van zijn berekeningen voor als een grafe, en leidt daaruit een aantal lokale beslissingspunten af (namelijk één voor elke pijl in de grafe: de vroegst mogelijke starttijd voor de eindknoop van de pijl, tenminste voor wat betreft de beperking uitgedrukt door die pijl). Daardoor wordt het mogelijk de globale performantie uit te drukken in functie van lokale beslissingspunten. Het is dan mogelijk om voor elke pijl de partiële afgeleide van de globale performantie naar de vroegst mogelijke starttijd voor die pijl off line te berekenen en door te geven aan de on-line besturing.

Het on-line-besturingsholon zal dan op zijn beurt dat advies gebruiken. Voor elk beslissingspunt zal het een aantal alternatieven vooropstellen. Het berekent de impact van elk alternatief op de vroegst mogelijke starttijden van elke relevante pijl in de grafe. Het is dan mogelijk om de impact van elk alternatief op de globale performantie te schatten door linearisatie (De afwijking in performantie wordt geschat door de som van de producten van de afwijking in vroegst mogelijke starttijd met de bijhorende partiële afgeleide). De verschillende alternatieven worden geëvalueerd op basis van hun verbetering of verslechtering van de globale performantie.

Men kan zich afvragen of het zin heeft om een dergelijk niet-lineair systeem te lineariseren. Echter, de bestaande alternatieven om een planning uit te voeren zijn zelfs helemaal niet gebaseerd op één of andere reflectie naar performantie toe. Bovendien wordt de on-line besturing niet volledig aan haar lot overgelaten als er storingen komen: er wordt ondertussen een nieuwe planning berekend, maar zolang die niet gereed is, moet de on-line besturing verder werken met de bestaande planning. Het gebruik van perturbatie-analyse is dus enkel nodig voor kortetermijnbeslissingen. De experimenten in hoofdstuk 8 bevestigen trouwens dat de resultaten best bevredigend zijn.

## 6.7 Besluit

Dit hoofdstuk heeft drie heuristische planningsuitvoeringsalgoritmen en één theoretisch gefundeerd algoritme voorgesteld. Een belangrijk aspect van holonische productiebesturing is dat het algoritme niet opgelegd wordt door de architectuur, en bijgevolg dat er gemakkelijk tussen algoritmen kan gewisseld worden. Dit komt ook tot uiting in de implementaties.

## 7. Ontwerp en implementaties

De concepten en algoritmen ontwikkeld voor deze thesis werden geïmplementeerd in een onderzoeksprototype-besturing voor een flexibel assemblagesysteem. Dit prototype bestaat uit een reactieve fijnplanner PaRSiFAL, een on-line-besturingsholon PHOCS en een aantal lokale holons (werkstationholons, orderholons en een transportsysteemholon). Elk holon is geïmplementeerd als een agent (een apart computerproces), elk met een eigen gebruikersinterface. De holons communiceren met elkaar via een asynchroon boodschap-gebaseerd protocol over ethernet.

Het testbed is een flexibel assemblagesysteem met zes werkstations, die in willekeurige volgorde te bezoeken zijn. Het productiesysteem heeft dus de lay-out van een *job-shop*. De werkstations zijn verbonden via een flexibel transportsysteem, dat de orders transporteert via lopende banden op paletten.

PaRSiFAL bevat een aantal planningsalgoritmen. Enkele zijn eenvoudige dispatchingregels (FIFO, EDD en aanvullingen daarop). Er zijn ook enkele omgevingszoekalgoritmen die via de eerst gevonden dalende helling op zoek gaan naar een lokaal optimum. Er zijn ook algoritmes die de afdalingsmethode toepassen voor verschillende startoplossingen en algoritmes die uit een lokaal optimum proberen ontsnappen door de structuur van de oplossing te herorganiseren.

PHOCS bestuurt het productiesysteem door de paletten te beheren, transportverzoeken uit te sturen, *set-ups* te commanderen en operaties te starten. Op die manier maakt hij beslissingen over het toekennen van operaties aan machines en het sequensen daarvan op die machines. In PHOCS kan men via een menu de productie manueel (interactief) sturen. Men kan ook kiezen uit een reeks besturingsalgoritmen (zoals beschreven in hoofdstuk 6) om de productie te runnen.

## 8. Experimenten

Dit hoofdstuk beschrijft de experimenten die uitgevoerd werden om de nieuwe, holonische benadering voor productiebesturing te toetsen aan hiërarchische en heterarchische besturing. Daarom wordt de performantie van de verschillende algoritmen (hiërarchische, heterarchische, en de vier holonische algoritmen) vergeleken voor een aantal testproblemen onder een aantal scenario's. Die scenario's verwijzen zowel naar productieomstandigheden zonder storingen als naar omstandigheden met modelfouten, stochastische afwijkingen van de geplande duur en machine-uitval. Het doel van die experimenten is niet alleen het vergelijken van de resulterende systeem-

## Samenvatting

---

performantie met en zonder storingen, maar ook de vergelijking van de voorspelbaarheid van al die benaderingen.

Eerst werd nagegaan hoe repeteerbaar experimenten zijn. Voor een experiment van ongeveer 1000 s werd voor hiërarchische besturing een repeteerbaarheid van 2 s (standaardafwijking) opgemeten. Wel kan die repeteerbaarheid verslechteren (tot een kleine 10 s) als het optimalisatiegedrag van de fijnplanner stochastische effecten vertoont. Voor heterarchische systemen is de voorspelbaarheid veel slechter (25 s voor de gemiddelde performantie en 70 s voor de gemiddelde standaarddeviatie van een individueel order). Holonische systemen (als ze zich naar behoren gedragen) zitten daar ergens tussenin, met een voorspelbaarheid van 10 s voor de gemiddelde performantie tot 25 s voor individuele orders.

disturbance scenario.	Hierarchical	STA (Hol.)	AID (Hol.)	LDFS (Hol.)	PA (Hol.)	Heterarchical
NoDist	336 s	366	—	—	370	556
ModelErr	416	423	494	520	443	556
BrkDwn3	468	472	534	533	483	580
BrkDwn2	527	442	574	565	468	602
BrkDwn1	645	612	640	672	590	695

**Legend:** The table shows  $\bar{T}^w$ , for problem large1.dat, under a number of disturbance scenarios and for a number of control algorithms. Single scheduling run.

Tabel 1. Vergelijking van de vier holonische besturingsalgoritmen met hiërarchische en heterarchische besturing. Voor de verklaring van gebruikte terminologie wordt verwezen naar de Engelstalige tekst.

De resultaten van de vergelijkende experimenten voor wat betreft de laattijdigheid zijn weergegeven in tabel 1. Hieruit blijkt dat hiërarchische systemen, als ze behoorlijk geoptimaliseerd worden, een betere performantie hebben dan heterarchische indien er geen storingen optreden. Heterarchische besturing is echter ongevoeliger voor storingen. Holonische algoritmen hebben wel degelijk een potentieel om zowel hiërarchische als heterarchische systemen te overtreffen in performantie. Als de storingen groter worden, zijn de STA-heuristiek en het PA-algoritme beter dan zowel hiërarchische als heterarchische besturing. Wel is het duidelijk dat niet alle holonische algoritmen, zeker de heuristische niet, in alle omstandigheden beter presteren dan de klassieke aanpak. Het algoritme op basis van perturbatie-analyse lijkt minder goed te presteren dan verwacht, maar dat is voor een groot deel te wijten aan het feit dat de computer overbelast is voor dit algoritme. Er is zeker nog verder

onderzoek nodig om de performantie van deze systemen verder te verbeteren en om de betrouwbaarheid van de experimenten te verhogen.

Deze experimenten tonen aan dat er zeker voordeel te halen is uit het invoeren van hiërarchie in gedistribueerde productiebesturing. Ook het verdelen van de beslissingsbevoegdheid over verschillende holons biedt een betere performantie dan afwisselend (afhankelijk van de toestand van het systeem) hiërarchisch en heterarchische te gaan werken.

Het gebruik van gelijktijdige fijnplanning en planningsuitvoering werd reeds in eerdere hoofdstukken geanalyseerd wat betreft de performantie. Deze analyse was echter gebaseerd op enkele vereenvoudigende aannamen. Experimenten hebben aangetoond dat deze aannamen realistisch waren.

## 9. Besluit

Het optreden van storingen is een van de belangrijke belemmeringen om succesvol gebruik te maken van optimaliserende fijnplanners in de industrie. Het concept van holonische productiesystemen is een veelbelovend paradigma om deze problematiek aan te pakken. Door zich te baseren op dit paradigma, levert deze thesis een aantal bijdragen op het gebied van holonische productiebesturing. Een aantal concrete concepten zijn voorgesteld om robuustheid tegen storingen te combineren met een hoge performantie. Er werden een besturingsarchitectuur en een aantal besturingsalgoritmen ontwikkeld om die concepten te realiseren. Implementaties en experimenten hebben het potentieel van holonische productiebesturing aangetoond.

Het onderzoek naar holonische besturingsalgoritmen is echter nog maar net gestart, en er is zeker nog vraag naar nieuwe en betere algoritmen. Er blijven trouwens nog heel wat problemen over met het fijnregelen van de parameters van die algoritmen. De zoektocht naar zelfregelende systemen belooft dus nog interessant te worden. Een ander fundamenteel probleem van autonome systemen is hun onvoorspelbaar gedrag, dat we eigenlijk nauwelijks begrijpen, en zeker nog niet onder controle kunnen houden.

Toch denk ik dat de tijd rijp is om de ontwikkelde concepten industrieel toe te passen. Productiebesturing is een snel groeiende markt, en vanuit de industrie bestaat er duidelijk vraag naar verdere onderzoek en ontwikkeling van geavanceerde werkvloerbesturingen.

## Samenvatting

---