

## **ARQUITETURA DE CONTROLE DISTRIBUIDO PARA SISTEMAS PRODUTIVOS**

**Caio Cesar Fattori, paicaio@usp.br<sup>1</sup>**  
**Samira Souit, samirasouit@gmail.com<sup>1</sup>**  
**José Isidro Garcia Melo, josgarme@univalle.edu.co<sup>2</sup>**  
**Fabrcio Junqueira, fabri@usp.br<sup>1</sup>**  
**Paulo Eigi Miyagi, pemiya@usp.br<sup>1</sup>**

<sup>1</sup>Escola Politécnica da Universidade de São Paulo, Brasil

<sup>2</sup>Universidad del Valle, Colombia

**Resumo.** *Os mercados estão se tornando independentes de barreiras geográficas e as indústrias têm procurado novas configurações de sistemas produtivos (SPs), passando de estruturas centralizadas para estruturas distribuídas, deslocando suas plantas produtivas para países com reservas de energia e baixos custos operacionais. Para permitir a coordenação e gerenciamento desta nova estrutura distribuída, elas aproveitam os avanços das tecnologias mecatrônicas e da informação, as quais permitem uma maior cooperação transnacional e provêm aos administradores e engenheiros uma visão global dos SPs. Neste novo contexto, as empresas são consideradas partes de um consórcio onde os SPs são compostos por componentes (subsistemas) com diferentes níveis de autonomia. Estes componentes independentes são instalados de modo disperso fisicamente e, para o usuário do sistema, formam uma unidade. Desta forma, um componente do sistema distribuído pode compartilhar os recursos dos outros componentes, o que assegura uma flexibilidade estrutural. Esse tipo de sistema apresenta novos problemas de integração e coordenação de componentes, que têm que ser superados para se chegar a uma efetiva implementação. A implementação de um SP distribuído e disperso deve assegurar a flexibilidade dos processos produtivos, os procedimentos de rearranjo de processos para atender a futuros requisitos, a escalabilidade do sistema permitindo a inclusão de novos componentes sem reduzir o desempenho, a padronização das partes estruturais de modo a manterem a integração em um ambiente heterogêneo, próprio dos SPs, e a integridade da informação e o pronto acesso aos operadores do sistema. Estes problemas indicam que o controle dos processos produtivos deve garantir uma troca de mensagens de comando e de monitoração com segurança considerando a estrutura distribuída e dispersa do SP. Desta forma, este trabalho propõe uma arquitetura de controle distribuído, orientada a serviço para a supervisão dos sistemas produtivos.*

**Palavras-chave:** *sistemas produtivos, sistemas distribuídos, orientação a serviço, arquitetura de controle*

### **1. INTRODUÇÃO**

Desde a década de 1990, os sistemas de produção e de cadeia de suprimentos têm alterado sua forma de produção da tradicional produção em massa liderada pelo produto para a customização em massa, causada pelo aumento da competição do mercado global (Thomas and Artiba, 2009). Com mercados mais exigentes para os produtos de alta qualidade a custos mais baixos, altamente customizados e com ciclos de vida curtos, alguns requisitos tornaram-se mais importantes, como qualidade e tempo de resposta, e outros novos foram impostos, como flexibilidade estrutural e integração na execução de processos produtivos mais abrangentes, visando manter a competitividade da empresa no mercado internacional. Neste tipo de mercado, as empresas não devem atuar independente, sendo forçadas a reconsiderar a maneira como elas se organizam para aumentar sua competitividade ( Nassar *et al.* (2008) ; Leitao (2009) ). Consequentemente, a pesquisa em controle de sistemas produtivos (SPs) mudou da abordagem tradicional centralizada para arquiteturas distribuídas e empresas espalhadas por todo o mundo (Junqueira *et al.*, 2005), tornado-se um sistema distribuído. Um dos fatores que contribuiu para a descentralização da produção é o deslocamento dos SPs para locais em que os custos de energia e operação são menores (Shah, 2005). O crescimento da complexidade dos novos produtos também contribuiu com a descentralização da produção por fazer com que as empresas de manufatura precisem colaborar, umas com as outras, na produção, na cadeia de suprimentos ou em redes de serviços, em que cada parceiro na cadeia ou na rede contribui com parte do produto final (Grefen *et al.*, 2009). Um sistema distribuído é uma coleção de entidades autônomas, chamadas de nós do sistema. Para minimizar a complexidade na implementação de um sistema distribuído, uma aplicação *middleware* é utilizada para

interligar os nós do sistema. E desta forma, permitir a integração das atividades nos nós e o compartilhamento dos recursos distribuídos (Bondavalli *et al.*, 2010).

No início da década de 1990, muitos SPs adotaram a estratégia de produção enxuta para aumentar a sua competitividade global. Essa estratégia é derivada do conceito de *Just-In-Time*, desenvolvida pela Toyota na década de 1960, e propõe que os SPs minimizem o consumo de recursos em todas as etapas de produção. Essa estratégia induzia a redução da flexibilidade do SP, pois quanto mais rígidos os processos maior a eficiência atingida (Lasa *et al.*, 2009). Com o crescimento da tecnologia da informação (TI) e da mecatrônica, tem-se procurado uma maior eficiência na produção sem aumentar a rigidez dos SPs (Bruun and Mefford, 2004). Nesse contexto, a pesquisa de SPs tornou-os capazes de produzir com maior diversidade e eficiência. Máquinas de comando numérico como *computadorized numerical control* (CNC) e robôs industriais permitiram uma produção flexível e eficiente. As tecnologias auxiliadas por computador, como o CAD/CAM, foram integrados aos SPs criando os sistemas de manufatura flexíveis (FMS - *Flexible Manufacturing System*) e os *computer integrated manufacturing* (CIM). A exemplo dos FMS, o projeto de sistemas de controle é um campo amplamente discutido na comunidade científica. Os múltiplos e, por vezes, contraditórios requisitos de produtividade, qualidade e diversificação de produtos influenciam o projeto do FMS e seu controle. Um FMS é um SP reprogramável, operando com processamento de dados distribuídos e fluxo de materiais automatizado usando uma arquitetura de controle retroalimentada altamente flexível, controlada por computador (Ausfelder *et al.*, 1994).

Em geral, considera-se que as empresas estão apoiadas em 4 processos de negócios: processo de desenvolvimento de produtos, processo de atendimento de pedidos, processo de atendimento ao cliente e processo de aquisição de clientes. Além disso, o aumento da velocidade de transmissão da *internet* permite o aprimoramento desses processos pelas empresas, especialmente nos que se referem aos clientes. Quando aplicados (esses processos) em um sistema produtivo distribuído, surgem novos processos do ponto de vista gerencial, como o processo de gerenciamento de fornecedores e processo de gerenciamento de transporte de materiais, que se tornam mais relevantes quando as etapas do processo produtivo ocorrem fisicamente em diferentes locais (Luo *et al.*, 2003). Nesse tipo de SP, o processo produtivo deve ser integrado. Para isso, os SPs devem coordenar suas operações de modo que haja sincronização das etapas do processo produtivo. Em alguns trabalhos, a coordenação de tarefas é aplicada a sistemas distribuídos e a composição de tarefas coordenadas gera um fluxo de dados entre os nós (ou SPs) envolvidos, modelados como uma rede de processos (Palomera-Perez and Benitez-Perez, 2009).

A integração dos SPs tem alguns requisitos como (Cheng *et al.*, 2010):

- **Facilidade instalação e configuração.** Inclui 3 (requisitos) que são: acomodar os membros com uma variedade de graus de sofisticação de TI, oferecer uma ampla faixa de funcionalidades e permitir uma constante mudança da lista de empresas e clientes. Em especial, o terceiro requisito faz com que a integração de sistemas seja flexível, permitindo uma instalação rápida de novos elementos ao sistema;
- **Baixo Custo.** A maior parte das empresas utiliza poucos funcionários e é pequena ou média. Para viabilizar a participação dessas, o custo das aplicações de *software* para integração dos sistemas não pode ser elevado;
- **Facilidade de conexão e integração.** Esse requisito interfere diretamente na variedade de graus de sofisticação de TI. Ele se deve ao fato de que, em alguns ramos de atuação, os usuários se diferenciam em experiências em níveis educacionais prévios. Isso significa que a integração deve oferecer um ambiente com diferentes níveis de complexidade para atender dos usuários menos experientes aos mais experientes;
- **Capacidade de integrar sistemas externos e de informação.** Com sistemas distribuídos geograficamente, cada um pode usar e manter suas informações separadamente. Para esses, não só é desejável expor suas aplicações internas para sistemas externos como é benéfico permitir a integração e a conexão com esses, por troca de funções e informações, em um projeto colaborativo;
- **Acesso personalizável a informações e aplicações.** Alguns participantes podem ser relutantes ao compartilhamento de informações com outros que não possuem relações comerciais com estes. A integração deve permitir ao usuário personalizar o acesso a informação e aplicações que este oferece.

Os SPs estão evoluindo para ambientes mais ágeis para atender as constantes mudanças de requisitos dos clientes e do ambiente de mercado. Desta forma, tem crescido o número de empresas com foco apenas em suas principais competências, de modo que agora dependem de outras para prover especialidades complementares e recursos. A colaboração na manufatura faz com que empresas, clientes e parceiros trabalhem juntos para obter maior ganho e, neste contexto, a orientação a serviços tem atraído a atenção de pesquisadores (Yan *et al.*, 2010).

Este trabalho propõe uma arquitetura de controle que seja capaz de teleoperar sistemas distribuídos, considerando que nesses sistemas as informações encontram-se dispersas e onde cada elemento desse sistema deve ser independente dos demais. Essa arquitetura usa o conceito de orientação a serviços com *Web Services* (WS).

A estrutura deste trabalho apresenta na seção II uma definição geral da arquitetura com orientação a serviços. A seguir, é apresentada uma proposta de uma arquitetura de controle distribuído para controle de sistemas produtivos. Na próxima seção, é descrito um exemplo de aplicação. Finalmente, são apresentadas as conclusões do trabalho.

## 2. SERVICE ORIENTED ARCHITECTURE

A *Service-oriented Architecture* (SOA) é um paradigma na área da tecnologia da informação (TI) para organizar e utilizar capacidades computacionais (como processamento, armazenamento, filtragem e outras) distribuídas de uma forma modular e que podem ser implementadas independentemente da tecnologia. Em geral, num ambiente distribuído, é normal que as necessidades de uma aplicação computacional sejam atendidas pelas capacidades oferecidas por outra aplicação computacional. Neste contexto, os requisitos de um agente computacional de certo proprietário podem ser atendidos por outro agente computacional pertencente a outro proprietário. Assim, não há necessariamente uma correlação de um para um entre necessidades e capacidades, ou seja, uma capacidade não é feita especificamente para atender uma necessidade, mesmo que seja usada para atender essa. A granularidade das necessidades e capacidades varia da fundamental à complexa, já que dependem da quantidade de operações computacionais feitas dentro de uma capacidade ou da quantidade de operações computacionais requeridas dentro de uma necessidade, e uma necessidade pode requerer a combinação de várias capacidades enquanto uma simples capacidade pode atender mais do que uma necessidade (OASIS (2006) ; Wang *et al.* (2009) ). Com o conceito de serviço, que é a implementação de uma capacidade computacional, um sistema de informação pode ser melhorado para um novo nível e ser mais adequado a ambientes interoperáveis e heterogêneos (Wang and Hu, 2009).

A (arquitetura) SOA provê um poderoso *framework*<sup>1</sup> para encontrar necessidades e capacidades e para combinar capacidades para atender uma necessidade ( OASIS (2006) ; (Zhu and Melo, 2009)). SOA pode ser considerada uma abordagem de integração universal alavancando funcionalidades de *softwares* existentes. A lógica de integração de orientação a processo é separada da interação do usuário, ou seja, a interface com o usuário é feita de uma forma independente da forma como é feito o controle de atividades. A conectividade é acoplada pelos adaptadores, ou seja, um serviço comunica-se com outro por um adaptador. Um *message broker* (corretor de mensagens) fornece um mecanismo para validação, transformação e filtro de dados para a SOA (Mayerl *et al.*, 2008).

Com as características acima, a SOA adota um tipo de acoplamento considerado "fraco", ou seja, a comunicação entre serviços independe da forma como esses foram criados, mas de adaptadores ( Shan *et al.* (2009) ; Mayerl *et al.* (2008) ; Zhu and Melo (2009) ; Wang and Hu (2009) ) e usa coleções de serviços reusáveis que podem ser recombinaados para definir novos processos ou refinar um já existente. A idéia fundamental é similar a de modelos baseados em agentes: um serviço de baixo nível pode ser parte de alguns serviços de alto nível ou de processos de negócios colaborando com outros serviços enquanto tem certo grau de autonomia ( Medina *et al.* (2009) ; Wang and Hu (2009) ; Zhu and Melo (2009) ). Ambientes fracamente acoplados apresentam uma redução de dependências entre aplicações com interação de estilo de mensagem, semântica e sintaxe. Como é feita sobre outra abordagem de *middleware* (uma aplicação com ferramentas e interfaces disponíveis para comunicar-se com outras aplicações em outras camadas, por exemplo CORBA) existente, a implementação e as tecnologias específicas do detalhamento do serviço são ocultados atrás da definição de interfaces abstratas cedendo à forte autonomia, ou o fraco acoplamento, dos serviços a serem integrados (Mayerl *et al.*, 2008). Os serviços na SOA são auto-contidos, ou seja, executam tarefas pré-determinadas ( Zhu and Melo (2009) ; Wang and Hu (2009) ). A SOA provê um padrão de desenvolvimento flexível independente da complexidade da granularidade do serviço (Shan *et al.*, 2009).

Alguns conceitos descritos em OASIS (2006) relacionados com o paradigma SOA são:

- **Visibilidade:** refere-se ao poder das necessidades e das capacidades de ver uma as outras. É feita pela descrição de aspectos, como funcionais e requisitos técnicos; restrições e políticas relacionadas; e mecanismos de acesso e resposta.
- **Semântica:** refere-se a forma de como uma necessidade é descrita. A semântica e a sintaxe de uma necessidade devem ser amplamente acessíveis e entendíveis.
- **Interação:** é a atividade de usar uma capacidade. Tipicamente mediada pela troca de mensagens, uma interação prossegue por uma série de trocas de informações e chamadas de ação.
- **Contexto de execução:** é um conjunto de elementos técnicos e de negócios que fica entre as necessidades e capacidades. Este permite ao provedor (que fornece uma capacidade) e usuário (que tem uma necessidade de serviço) interagirem e estabelecerem um ponto de decisão para qualquer política e contrato que podem estar em vigor.

Os WSs foram concebidos para permitir o acesso consistente a recursos computacionais em múltiplas plataformas heterogêneas, suportando certo grau de colaboração inter-organizacional no gerenciamento de processos. WS estabelece

<sup>1</sup>Estrutura que dá suporte para poder trabalhar.

meios padronizados de inter-operação entre diferentes aplicações de *software* que podem ser executados em uma variedade de ambientes computacionais. Para integrar sistemas heterogêneos deve-se separar as definições da parte das interfaces da parte com protocolos de transferência particulares e formatos de codificação de dados (Kim *et al.*, 2006). Os padrões de WS são utilizados para sua especificação.

Segundo Han *et al.* (2008) para modelar um sistema baseado em WSs é necessário considerar os seguintes aspectos:

- Os modelos dos processos devem ter duas abordagens, uma lógica e outra física;
- O método de modelagem lógica do sistema baseado em WSs deve ser de fácil entendimento para o gerente de negócios;
- O método de modelagem lógica deve ser também intuitivo, visualizável e conveniente para ser usado por usuários, especialmente por gerentes e analistas de negócios;
- Deve possuir uma técnica efetiva de análise para avaliar o quão correto é o modelo lógico;
- O modelo lógico deve ser conveniente para ser transformado num modelo físico.

### 3. ARQUITETURA PROPOSTA

A arquitetura (estrutura lógica que fornece suporte aos elementos do sistema) de controle do SP disperso que está sendo considerado nesse trabalho está representado na Figura 1. Nesta arquitetura os elementos (atores) envolvidos são o "cliente", o "operador", o "sistema produtivo" e o "banco de dados", porém essa mesma arquitetura suporta múltiplos elementos de cada um desses. Esta arquitetura considera que os elementos podem estar dispersos geograficamente. Assim como especificado em SOA (Seção 2), cada elemento da arquitetura pode ser um usuário ou um servidor de um serviço disponibilizado para qualquer um dos elementos do sistema. O "cliente" utiliza o serviço "Registrar Pedido", disponibilizado pelo "banco de dados", para gravar as informações que serão usadas pelos "operadores" e pelos "sistemas produtivos". O "cliente" utiliza os serviços "Procura Operador" e "Procura SP", disponibilizados pelo "banco de dados", para obter uma lista de "operadores" e "sistemas produtivos", que oferecem serviços no sistema, e envia seu pedido a esses elementos, pelos serviços "Solicita Operador" e "Solicita SP" que são disponibilizados por eles, respectivamente. O "cliente" então é capaz de negociar com os "operadores" e os "sistemas produtivos" as condições de atendimento ao seu pedido como: custo, prazo de entrega, qualidade e outros. Após essa etapa, os "operadores" estão habilitados a alterar as informações escritas no "banco de dados", pelo serviço "Consulta/Altera Pedido", com novos dados que julgar necessário como: histórico de operações, porcentagem concluída do pedido, operador que atendeu a solicitação e outras. Por meio do serviço "Consulta/Altera Pedido", do "banco de dados", o operador pode também consultar as informações que para ele são relevantes para o atendimento do pedido como: quantidade de peças, sistema produtivo em que será feito o atendimento e outros. Os "sistemas produtivos" também são capazes de consultar e alterar as informações de um pedido pelo serviço "Consulta/Altera Pedido", do "banco de dados". Com os "operadores" e "sistemas produtivos" definidos, o pedido será atendido pelo "operador" utilizando o serviço "Opera SP" que é disponibilizado pelo "sistema produtivo".

Essa arquitetura considera um "cliente" dito experiente, ou seja, um "cliente" que tem conhecimento das atribuições necessárias dos "operadores" e dos "sistemas produtivos" para atender ao seu pedido. Os "operadores" e "sistemas produtivos" são responsáveis por organizar as suas respectivas filas de atendimento a múltiplos "clientes" e por garantir que as condições negociadas com cada "cliente" sejam atendidas, i.e., se um "operador" negocia com um "cliente" o atendimento de uma etapa de um pedido em um certo prazo esse passa a ser responsável pelo atendimento e pelo prazo acertado. Os "clientes" são responsáveis pela evolução do processo global que pode ser composto por uma ou mais etapas, i.e., o "cliente" deve negociar e acertar o início de cada etapa com um "operador" e um "sistema produtivo".

Os elementos "sistemas produtivos" possuem um controle local em que são especificadas as condições de operação, condições críticas de operação, estratégias de detecção de erros, estratégias de recuperação de falhas e outras funções de controle que independem dos elementos externos ao próprio.

Os elementos "sistemas produtivos" e "operadores" são efetivamente as plantas de manufatura (SPs) e os profissionais especializados (supervisores de processos produtivos), o serviço prestado por eles pode ter um tempo de resposta não determinístico. Essa condição deve ser prevista pelas estratégias de controle locais dos "sistemas produtivos" para garantir que uma requisição não aguarde, indeterminadamente, uma resposta do servidor.

### 4. EXEMPLO DE APLICAÇÃO

Um exemplo de SP distribuído é a planta de montagem de peças automatizada. Este sistema inclui subsistemas como: alimentação de peças, inspeção de peças, transporte de *pallets* e montagem de produtos, que estão interligados por meio de uma rede de comunicação, Fig. 2. O sub-sistema de alimentação executa o serviço de retirar peças de um *buffer*, e enviá-las para o sub-sistema de inspeção. O sub-sistema de inspeção executa o serviço de controle de qualidade e identificação das características físicas da peça. Este sub-sistema executa também o serviço de carregamento (em *pallet*)

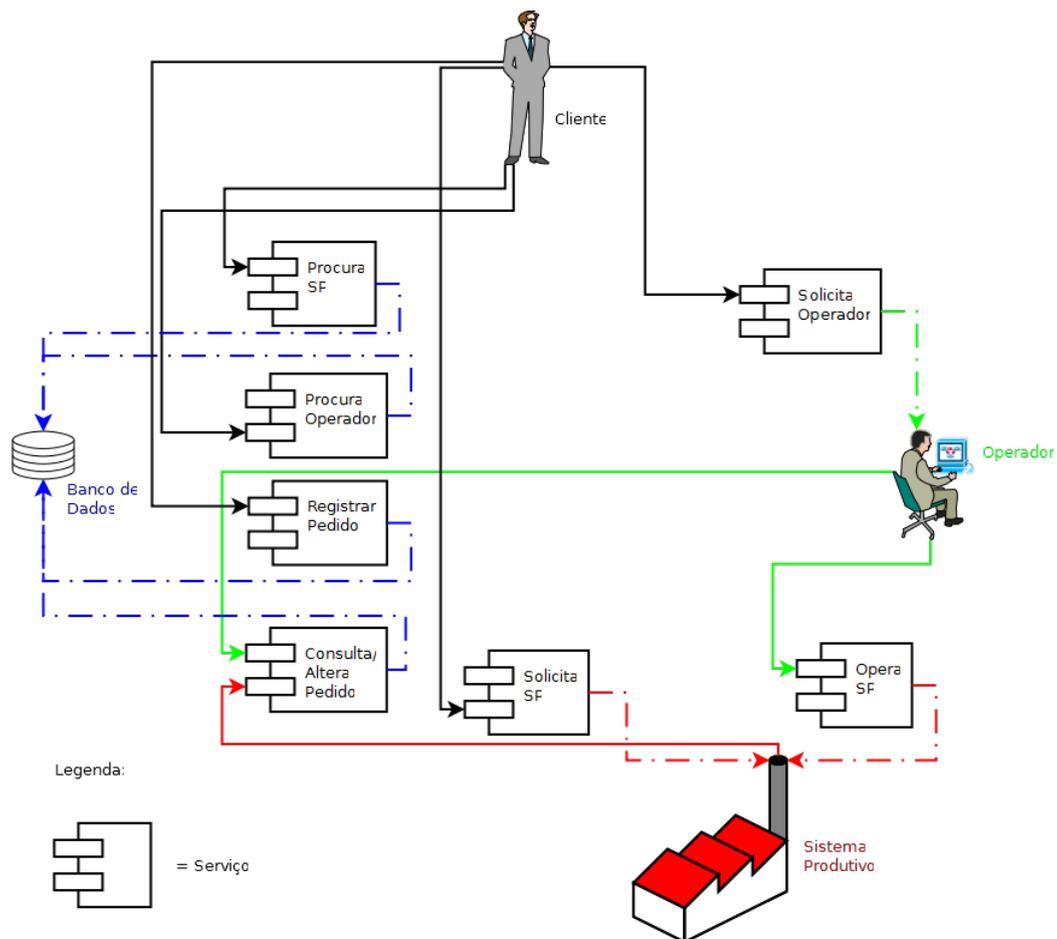


Figura 1. Arquitetura de controle de um SP disperso

da peça inspecionada. O subsistema de transporte executa o serviço de movimentação física de *pallets* entre os subsistemas de inspeção, montagem e retirada de produtos. O subsistema de montagem executa o serviço de montagem das peças.

No exemplo da Figura 2, num processo produtivo dividido em 4 atividades o usuário pode solicitar ao subsistema (que pode ser entendido como um SP) alimentação para fornecer matéria prima para a produção do seu pedido. Para garantir a execução correta dessa etapa, o usuário pode solicitar o operador op. alimentação para teleoperar o subsistema alimentação. O mesmo pode ser feito para os sub-sistemas inspeção, transporte e montagem, e para os operadores op. inspeção, op. transporte e op. montagem, respectivamente.

As solicitações de tipo e quantidade de produtos são definidas por um usuário, via *internet*. Todos os subsistemas podem ser monitorados e teleoperados também via *internet*. Os subsistemas disponibilizam, por meio de serviços, uma série de informações ao operador, para garantir o acompanhamento da execução remota do processo produtivo. Na teleoperação, além das informações de monitoração, uma série de comandos é disponibilizada ao operador para interagir com nos processos produtivos.

Esses subsistemas podem ser integrados utilizando a arquitetura apresentada na Figura 1 por meio da troca de informações entre os diferentes serviços executados nos subsistemas que integram a arquitetura, pode ser estabelecido um trabalho colaborativo envolvendo vários operadores e subsistemas com o intuito de produzir os produtos solicitados pelo usuário.

## 5. CONCLUSÃO

Essa arquitetura é proposta para um cliente dito experiente, ou seja, um cliente que tem conhecimento das atribuições necessárias dos operadores e sistemas produtivos para atender ao seu pedido. Com isso, não há um controle centralizado do sistema e todos os elementos dela podem ser arranjados numa mesma camada, (Fig. 1).

A arquitetura distribuída propõe que o cliente negocie todas as etapas do processo de fabricação, que podem ser feitas utilizando os mesmos operadores e sistemas produtivos ou diferentes a cada etapa. Após essa negociação o cliente possui uma agenda de processos.

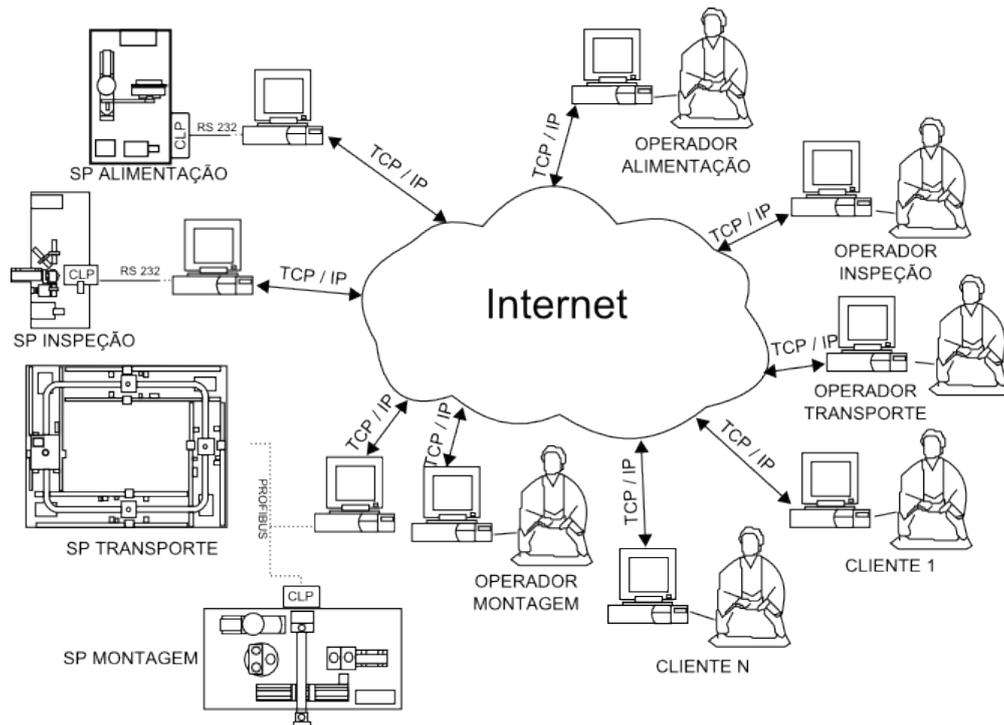


Figura 2. Exemplo de uso da arquitetura em um sistema distribuído

O serviço que o sistema produtivo disponibiliza, chamado Opera Fábrica, tem a implementação particular de cada prestador do serviço, já que esse serviço depende das configurações de equipamentos da planta. Neste caso, o prestador do serviço, sistema produtivo, deve fornecer capacidades suficientes para que um operador consiga teleoperar o sistema.

Como os elementos do sistema são independentes e oferecem ou requisitam serviços na camada de gerenciamento, eles são responsáveis pela organização de seus recursos, ou seja, se um sistema produtivo não tem matéria prima para atender ao pedido de um cliente ele não pode responder para o mesmo que é capaz de atender.

## 6. AGRADECIMENTOS

Os autores agradecem o apoio parcial do CNPq, CAPES e FAPESP para o desenvolvimento deste trabalho.

## 7. REFERENCIAS

- Ausfelder, C., Castelain, E. and Gentina, J.C., 1994. "A method for hierarchical modeling of the command of flexible manufacturing systems". In *IEEE Transactions on Systems, Man and Cybernetics*. IEEE, Vol. 24, pp. 564–573. ISSN 0018-9472.
- Bondavalli, A., Ceccarelli, A., Falai, L. and Vadursi, M., 2010. "A new approach and a related tool for dependability measurements on distributed systems". *IEEE Transactions on Instrumentation and Measurement*, , No. 99, pp. 1–12. ISSN 0018-9456. doi:10.1109/TIM.2009.2023815.
- Bruun, P. and Mefford, R.N., 2004. "Lean production and the internet". *International Journal of Production Economics*, Vol. 89, No. 3, pp. 247 – 260. ISSN 0925-5273. doi:10.1016/j.ijpe.2003.10.007. Manufacturing Strategy.
- Cheng, J.C., Law, K.H., Bjornsson, H., Jones, A. and Sriram, R., 2010. "A service oriented framework for construction supply chain integration". *Automation in Construction*, Vol. 19, No. 2, pp. 245 – 260. ISSN 0926-5805. doi: 10.1016/j.autcon.2009.10.003.
- Grefen, P., Mehandjiev, N., Kouvas, G., Weichhart, G. and Eshuis, R., 2009. "Dynamic business network process management in instant virtual enterprises". *Computers in Industry*, Vol. 60, pp. 86–103. doi:10.1016/j.compind.2008.06.006.
- Han, R., Liu, K., Ju, Y. and Zhao, J., 2008. "A Petri net theory-based method for modeling web service-based systems". In *4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08*. pp. 1–7. doi:10.1109/WiCom.2008.2839.
- Junqueira, F., Villani, E. and Miyagi, P.E., 2005. "A platform for distributed modeling and simulation of productive systems based on petri nets and object-oriented paradigm". In *10th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. Catania, Vol. 1, pp. 907–914. ISBN 978-078039402-5.
- Kim, W., Chung, M.J., Qureshi, K. and Choi, Y.K., 2006. "WSCPC: An architecture using semantic web services for

- collaborative product commerce”. *Computer in Industry*, Vol. 57, pp. 787–796. doi:10.1016/j.compind.2006.04.007.
- Lasa, I.S., de Castro, R. and Laburu, C.O., 2009. “Extent of the use of lean concepts proposed for a value stream mapping application”. *Production Planning & Control: The Management of Operations*, Vol. 20, No. 1, pp. 82–98. ISSN 0953-7287. doi:10.1080/09537280802685322.
- Leitao, P., 2009. “Agent-based distributed manufacturing control: A state-of-art survey”. *Engineering Applications of Artificial Intelligence*, Vol. 22, pp. 979–991. doi:10.1016/j.engappai.2008.09.005.
- Luo, R., Tzou, J.H. and Lan, C.C., 2003. “The development of web-based e-business system for rapid prototyping manufacturing”. *The 29th Annual Conference of the IEEE Industrial Electronics Society, 2003. IECON '03.*, Vol. 2, pp. 1290 – 1295. doi:10.1109/IECON.2003.1280239.
- Mayerl, C., Vogel, T. and Abeck, S., 2008. “Soa-based integration of service management applications soa-based integration of service management applications integrated it service management”.
- Medina, I., Garcia-Dominguez, A., Aguayo, F., Sevilla, L. and Marcos, M., 2009. “Proposal of a methodology for implementing a service-oriented architecture in distributed manufacturing systems”. In V.J. Segui, ed., *Third Manufacturing Engineering Society International Conference: MESIC-09*. AIP, Vol. 1181, pp. 622–632. doi:10.1063/1.3273683.
- Nassar, M.G.V., Garcia Melo, J.I., Miyagi, P.E. and dos Santos Filho, D., 2008. “Modeling and analysing of the material entry flow system in a pickling line process using Petri nets”. In *ABCMS Symposium Series in Mechatronics*. Vol. 3, pp. 444–453.
- OASIS, 2006. *Reference Model for Service Oriented Architecture 1.0*.
- Palomera-Perez, M. and Benitez-Perez, H., 2009. “Scheduling coordinated task”. In *4th IEEE Conference on Industrial Electronics and Applications, 2009. ICIEA 2009*. pp. 3944 –3949. doi:10.1109/ICIEA.2009.5138947.
- Shah, N., 2005. “Process industry supply chains: Advances and challenges”. *Computers & Chemical Engineering*, Vol. 29, No. 6, pp. 1225 – 1236. ISSN 0098-1354. doi:10.1016/j.compchemeng.2005.02.023.
- Shan, J., Ma, D., Zhang, B. and Luo, W., 2009. “Research and implementation of code generator for information system based on SOA”. In *International Conference on Scalable Computing and Communications; Eighth International Conference on Embedded Computing, 2009. SCALCOM-EMBEDDEDCOM'09*. pp. 143–147. doi:10.1109/EmbeddedCom-ScalCom.2009.34.
- Thomas, A. and Artiba, A., 2009. “Modelling and control of productive system: Concepts and application”. *International Journal of Production Economics*, Vol. 121, No. 1, pp. 1 – 3. ISSN 0925-5273. doi:10.1016/j.ijpe.2009.04.002.
- Wang, H. and Hu, Q., 2009. “Research and application of an integration platform for e-commerce system based on SOA”. In *International Conference on Management of e-Commerce and e-Government, 2009. ICMECG '09*. pp. 424–427. doi:10.1109/ICMeCG.2009.113.
- Wang, J., Song, Y.T. and Xiong, Y., 2009. “Service evaluation in SOA: Toward business/IT alignment”. In *10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, 2009. SNPD '09*. pp. 310–315. doi:10.1109/SNPD.2009.105.
- Yan, J., Ye, K., Wang, H. and Hua, Z., 2010. “Ontology of collaborative manufacturing: Alignment of service-oriented framework with service-dominant logic”. *Expert Systems with Applications*, Vol. 37, No. 3, pp. 2222 – 2231. ISSN 0957-4174. doi:10.1016/j.eswa.2009.07.051.
- Zhu, W. and Melo, W., 2009. “Refactoring J2EE application for JBI-based ESB: A case study”. In *IEEE International Enterprise Distributed Object Computing Conference, EDOC*. pp. 213–217. ISSN 1541-7719. doi:10.1109/EDOC.2009.25.

## 8. DIREITOS AUTORAIS

Os autores são os únicos responsáveis pelo conteúdo do material impresso incluído no seu trabalho.