

A dinâmica e as suas propriedades dos ambientes de desenvolvimento distribuído de software: um estudo de caso

Alexandre L’Erario

Engenharia de Produção - Escola Politécnica da Universidade de São Paulo - USP
São Paulo, Estado de São Paulo, CEP 05508-900, Brasil
Universidade Tecnológica Federal do Paraná – Campus Cornélio Procópio
Cornélio Procópio, Paraná CEP 86300-000, Brasil
Fundação Educacional do Município de Assis
Assis, Estado de São Paulo, CEP: 19807-634, Brasil
alexandre.lerario@femanet.com.br

e

Marcelo Schneck de Paula Pessoa

Engenharia de Produção - Escola Politécnica da Universidade de São Paulo - USP
São Paulo, Estado de São Paulo, CEP 05508-900, Brasil
mpessoa@usp.br

Abstract

The software distributed development (SDD) can reduce the time and the production cost for this reason many organizations join in this modality. The quantity of these partnerships is more e and more growing. The SDD environment has a group of properties. This article presents some common properties of these environments that were identified from a case study and of a network dynamics model, considering the coordination mechanisms. The mapping discussed in this research helps the most effective network management because it identifies common points that can reveal trouble and it explains a general positioning of the network.

Keywords: software distributed development, SDD, SDD environment dynamics.

Resumo

O desenvolvimento distribuído de software (DDS) pode reduzir o tempo e o custo de produção, por isso a quantidade de organizações que ingressam nesta modalidade, estabelecendo parcerias, é cada vez mais crescente. Os ambientes de DDS carregam consigo um conjunto de propriedades. Este artigo apresenta algumas propriedades comuns destes ambientes que foram identificadas a partir de um estudo de caso e de um modelo de dinâmica da rede associados aos mecanismos de coordenação. O trabalho de mapeamento discutido nesta pesquisa auxilia o gerenciamento mais eficaz da rede uma vez que identifica pontos comuns que podem revelar impasses e esclarecer um posicionamento geral da rede.

Palavras chave: Desenvolvimento distribuído de software, DDS, dinâmica de ambientes de DDS.

1. INTRODUÇÃO

Novas necessidades emergentes de mercado fazem com que as organizações que produzem software estabeleçam parcerias com o intuito de produzi-los. Uma maneira de estabelecer parceria entre as empresas que produzem software é construir uma rede de produção. Segundo Alstynne [1] uma rede de organizações pode ser definida pelos elementos envolvidos na estrutura da rede, processo e propósito. No entanto, estas parcerias nem sempre são locais, ou seja, são compostas por várias organizações distantes fisicamente umas das outras. O resultado, segundo Herbsleb [2] é a criação de organizações virtuais ou sistemas de times virtuais que podem operar com sucesso sobre as barreiras geográficas e culturais.

O desenvolvimento global de software tem tomado grandes proporções, pois, as organizações das mais diversas categorias descobrem constantemente que o desenvolvimento distribuído de software pode aumentar a produtividade e reduzir o custo.

A utilização de computadores autônomos interligados em ambientes organizacionais favoreceu o surgimento de sistemas de trabalho em grupo, denominado segundo Antunes [3] como *groupware*. As mais diversas ferramentas de apoio a *groupware*, síncronas (serviço de mensagem instantânea) ou assíncronas (e-mail, fórum) passaram a ser utilizadas amplamente nas organizações. Contudo Grudin [4] dá provas de que o *groupware* pode limitar a cooperatividade dos participantes (fator social e motivacional; aspectos políticos do lugar de trabalho). Além disso, o desenvolvimento de atividades em grupo pode acarretar em conflitos que podem desencadear uma má execução das atividades e conseqüentemente comprometem o projeto e o produto final.

A identificação dos fatores que influenciam em um ambiente distribuído de produção de software oferece aos gerentes uma ferramenta que possibilita a compreensão do funcionamento da rede de produção como um todo. Além de oferecer um mapeamento do funcionamento, as propriedades embutidas em uma rede, incluem mecanismos de coordenação sócio-técnicos.

Este artigo mostra a dinâmica de um ambiente de desenvolvimento distribuído de software que se inicia pela concepção da rede e do projeto até o momento onde a organização desassocia-se da rede e o projeto é finalizado. A partir desta dinâmica de funcionamento, são identificadas algumas propriedades, que também são descritas por este artigo. Estas variáveis foram obtidas por meio de um estudo de caso, onde foram analisados dois projetos desenvolvidos distribuídamente.

2. DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

O desenvolvimento distribuído de software (DDS) ocorre quando vários nós/sites (unidade independente de produção de software distante das demais) cooperam e/ou colaboram para desenvolver um mesmo produto ou parte dele. Neste cenário, a complexidade do processo de desenvolvimento se amplia. A redução do tempo é a principal razão da divisão de tarefas na produção distribuída de software, porém o tempo de comunicação e de resposta entre os nós pode ser incomensurável. Segundo Herbsleb [2] e Suzuki [5], há muitas variáveis neste cenário, tais como a cultura, a língua, a capacidade de cada nó entre outras. Além disso, Becker [6] afirma que desenvolver software em uma rede de produção requer gerenciamento mais eficaz de tarefas. Martin [7] afirma também que este gerenciamento deve estender-se a ferramentas, capacidades e informação.

Em sua pesquisa, Sa [8], afirma que o crescimento da internet e de ferramentas de CSCW influenciaram diretamente e, principalmente na difusão do desenvolvimento global de software, porém, embora a internet ofereça inúmeras vantagens em relação à comunicação, um projeto de software é muito complexo e grande. Por este motivo Suzuki (1999) afirma a necessidade de um

framework que deve esclarecer como a informação é gerenciada na rede e como as tarefas são delegadas, por exemplo.

Empresas que distribuem filiais ou efetuam *offshore* e/ou *nearshore* com o objetivo de aumentar a produtividade, reduzir o custo e tornar mais compatível com características locais são organizações que aderem ao GSD e segundo Carmel [9], o número delas está crescendo. Todavia dividir a produção de um software entre diversos nós envolve uma série de problemas. Alguns deles que localmente são insignificantes ganham proporção quando uma tarefa é dividida entre organizações fisicamente distantes.

3. A DINÂMICA DE UM AMBIENTE DE DDS

A dinâmica de um ambiente de DDS revela o funcionamento da rede, da sua concepção inicial até seu encerramento. Abordar individualmente os *sites* na rede pode não revelar a influência que o projeto tem sobre sua existência na rede, por isso, a dinâmica da rede, representada na figura 1, aborda conjuntamente dois componentes fundamentais em um ambiente de DDS. O primeiro componente indica a dinâmica de uma unidade de produção de software. O segundo elemento é atrelado ao projeto que é desenvolvido em uma rede de produção de software. Sobre esta dinâmica, há uma intersecção que ocorre quando o *site* precisa desenvolver um subproduto de rede, e há uma instância do projeto para vários *sites*.

As setas de transição na figura 1 indicam a transição dos estados do projeto e do *site*. Na figura 1 o elemento indicado como *Novo Projeto* indica a concepção de um novo produto de software por alguma organização, seja ela em rede ou não. A seta 1 da figura 1 indica que o plano gerencial do projeto em termos de componentização e divisão de tarefas foi realizado. Após esta seta, todos, ou parte dos artefatos estarão prontos para serem disponibilizados e distribuídos para que a rede e os *sites* possam desenvolvê-los. O elemento *pronto*, após a seta 1, indica que um papel de distribuidor de tarefas entra em ação para gerenciar a distribuição das tarefas do projeto na rede.

Na figura 1, o elemento *Novo site* indica que um novo nó da rede está pronto para associar-se a uma rede de produção de software. A elipse *pronto*, após a seta *a*, indica que o *site* já está na rede, ou seja, de alguma forma um acordo foi feito entre o *site* e a rede, e que pode ser, por exemplo, o conceito de matriz e filial. O estado de *pronto* após a seta *a* indica que a organização submeteu-se a alguma análise, resultando essa em uma agregação na rede. Neste estado o *site* está aguardando tarefas.

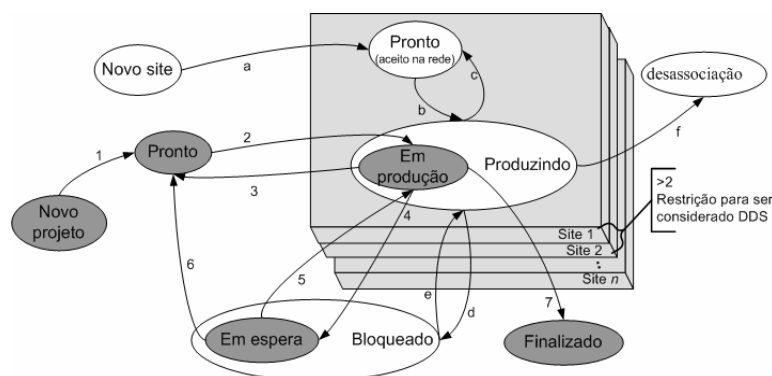


Figura 1 - A dinâmica de um ambiente de DDS

A seta 2 indica que o subproduto foi alocado em produção. Esta ação ocorre paralelamente à ação indicada pela seta *b*, que muda o estado do *site* de *pronto* para *produzindo*. O estado *Produzindo* efetua algumas tarefas de verificação, como por exemplo, aceitar ou não o trabalho. Se aceito, a tarefa entra em produção, caso contrário, a tarefa retorna e o papel do divisor de tarefas

precisará realocar o trabalho (seta 3). O *site* eventualmente fica ocioso, aguardando nova tarefa (seta *c*).

As setas 4 e *d* podem ocorrer quando há uma dependência de produtos de trabalho. Se isso ocorrer, para o projeto, o subproduto fica bloqueado, pois, por exemplo, não pode ser desenvolvido pelo fato de sofrer dependência funcional. O *site* fica ocioso, aguardando uma nova tomada de decisão. As seta 5 e *e*, representam respectivamente a resolução de dependência funcional e o retorno do nó na produção do artefato. A seta 7 indica quando o subproduto é finalizado e entregue à rede. Neste caso o estado *finalizado*, indica o fim da subtarefa para um determinado *site*. A ação indicada pela letra *f* indica a desassociação do *site* à rede que é removido da mesma e deixa de fazer parte do ambiente de DDS.

Na dinâmica, representada pela figura 1 há uma regra de restrição imposta sobre a quantidade de *sites* disponíveis. Para que seja considerado um ambiente de desenvolvimento distribuído de software, é necessário que a quantidade de *sites* seja maior ou igual a 2.

3.1. As propriedades dos ambientes de DDS

Há várias propriedades que diferem os ambientes de DDS. Cada propriedade determina o grau de complexidade de trabalho e revela consigo o quão difícil é de se produzir software em equipes fisicamente dispersas. Além disso, cada propriedade agrega consigo uma série de conceitos que são aplicados do nível estratégico ao operacional da rede.

Estas propriedades estão atreladas à dinâmica de funcionamento dos ambientes de DDS. Várias destas propriedades tornam-se eminentes, após determinados processos indicados na figura 1. A tabela 1 mostra a relação entre a dinâmica, baseada na figura 1, e as propriedades, abordadas neste capítulo. A primeira coluna desta mesma tabela indica qual elemento da figura 1, enquanto que a segunda coluna indica qual propriedade é instanciada.

Tabela 1 - A relação da dinâmica com as propriedades dos ambientes de DDS

Posição na figura 1	Propriedade eminente
Seta <i>a</i>	Mecanismo de coordenação
Seta <i>b, c</i>	Grau de interação
<i>Novo site</i>	Distância física e modelo de agrupamento
Seta 2, 3, <i>b, c</i>	Granularidade do repasse

A figura 2 ilustra as diversas as propriedades encontradas em um ambiente de DDS que são explanadas nas seções seguintes. Ela contém linhas escalares que compõem as diversas propriedades dos ambientes de DDS. A relação entre elas é expressa por uma condição mínima, definida na figura como: quantidade de *sites* maior que 1, distância, granularidade de repasse, difusão do processo e grau de interação maiores do que 0. Esta condição mínima indica o quanto de cada propriedade deve ser configurado para que o ambiente de desenvolvimento seja considerado distribuído.

Além disso, cada plano, composto por estas propriedades é orientado por um mecanismo de coordenação, indicado por Mintzberg [10]. A abrangência dos mecanismos de coordenação, na qual os ambientes de DDS utilizam englobam ajuste mútuo, padronização e supervisão direta e exercem influência contínua sobre as propriedades dos ambientes de desenvolvimento distribuído de software que define resumidamente como o processo de desenvolvimento distribuído ocorre. Por exemplo, se o mecanismo for ajuste mutuo, indica que ambos os *sites* tem a mesmo grau hierárquico e precisam definir ajustar mutuamente os artefatos trocados entre eles.

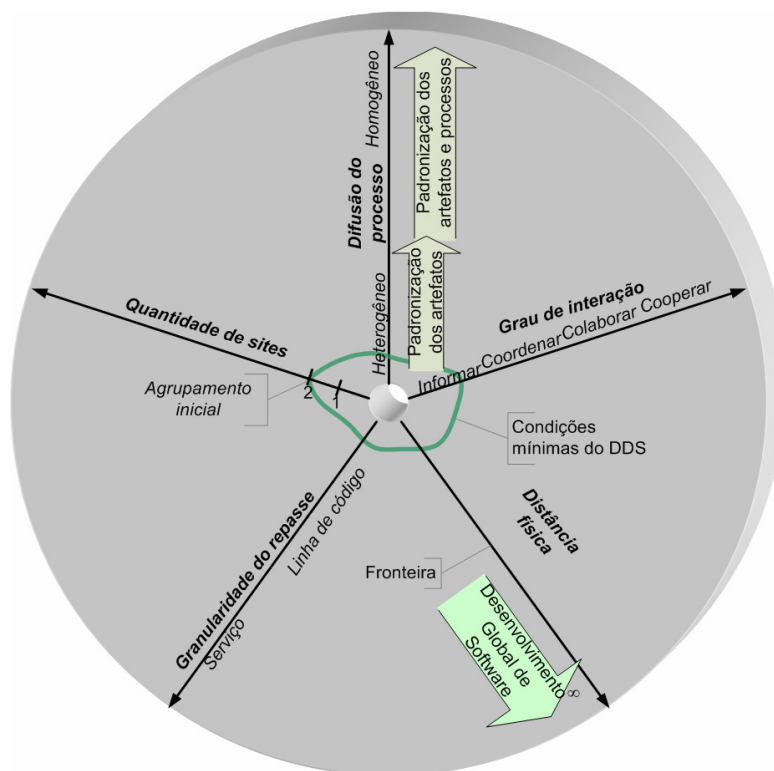


Figura 2 - Propriedades de um ambiente de DDS

O mecanismo de coordenação ajuste mútuo define que não há nenhuma padronização e os grupos precisam de ajustes contínuos entre artefatos trocados. A padronização como mecanismo de coordenação está subdividido em três categorias: padronização de resultados, processos e habilidades. A primeira subdivisão enquadra-se a padronização dos artefatos, a segunda do processo de software como um todo e a terceira, a padronização com relação a habilidades dos desenvolvedores existentes nos grupos. A supervisão direta indica total responsabilidade sobre um grupo, sendo que este deve definir como seus grupos subordinados trabalham.

As retas indicadas pela figura 2 são grandezas escalares. A reta definida como *difusão do processo* indica o quando o processo é homogêneo na rede. Nesta propriedade, a homogeneidade ou heterogeneidade do processo de todos os nodos do *site* são avaliadas. A reta definida como *grau de interação*, indica a intensidade de interação dos *sites*.

A reta definida como granularidade do repasse indica a forma na qual o artefato é encaminhado de um *site* para outro. Para ser considerado distribuído os nodos devem ficar há uma distância considerável, um do outro (indicado pela reta definida como *distância física* da figura 2). A última reta, definida como a *quantidade de sites*, neste caso deve ser maior do que 1 (um). As seções a seguir, explicam com mais detalhes o que cada propriedade aborda efetivamente dentro do ambiente de desenvolvimento distribuído de software.

3.1.1 Interação

A interação são estímulos enviados entre locais de produção com o objetivo de se realizar uma determinada tarefa. O grau de interação revela o quanto de informação é trocado entre os nodos e que neste caso há uma diferença de intensidade quando o software é desenvolvido de maneira colaborativa ou cooperativa.

No nível mais alto de interação, tem-se a cooperação, que é limitado por várias características, como metas em comum, plano compartilhado, processamento de dados para suporte

e cooperação coordenada. No nível de cooperação, metas do grupo têm prioridade sobre metas individuais. Em geral, essas decisões são baseadas no consenso do grupo. A cooperação é o desenvolvimento sobre o bem comum, por isso não é menos tendencioso a formação de hierarquias formalizadas, enquanto que a coordenação envolve diretamente a formação hierárquica. Borghoff [11] ainda, define onde é o posicionamento do desenvolvimento distribuído de software, que pode ocorrer por colaboração ou por cooperação.

O'Day [12] refere que enquanto na cooperação as relações de poder e os papéis dos participantes no trabalho cooperativo não são questionados. A colaboração envolve negociação cuidadosa, tomada conjunta de decisões, comunicação efetiva e aprendizagem mútua, num empreendimento que se foca na promoção do diálogo profissional. Pode-se ainda caracterizar a cooperação em síncrona e assíncrona. O primeiro caso envolve o trabalho simultâneo dos participantes, e o segundo, o trabalho não simultâneo [13]. Como exemplo, tem-se a vídeo-conferência e o uso de e-mail, respectivamente.

3.1.2 Distância geográfica

A distância geográfica está relacionada com o espaço físico entre os *sites* da rede. O Desenvolvimento Distribuído de Software, quando atinge proporções globais é chamado de Desenvolvimento Global de Software (GSD – *Global Software Development*) [14]. Pode-se considerar GSD como um tipo particular de DDS.

Shami [15] define “Desenvolvimento de software é crescentemente global, com desenvolvimento que ocorre em vários locais em regiões geograficamente diferentes”. Shami [15] afirma a importância das redes sociais em ambientes de DDS, que podem reduzir o impacto cultural sobre os *sites* da rede. Além disso, as redes sociais podem de certa forma, induzir a uma interação mais atraente entre os elementos da rede.

3.1.3 Difusão do processo

A difusão de processo está relacionada ao quão homogêneo a rede é em relação aos processos dos *sites*. A questão em relevância é mensurar o quanto do processo utilizado entre os *sites* é comum.

Para efetuar esta difusão, Mühlbacher [16] determinou um conjunto de objetos de negócio comum que os *sites* teriam que compreender. Cada objeto de negócio é composto por um ou mais workflow e contém regras de acesso. Além disso, criou o conceito de objetos móveis, que é constituída pela a informação que trafega entre os participantes. Para tanto, Mühlbacher [16] utiliza repositório centralizado com controle de versão.

Sobre uma análise mais formal, do ponto de vista matemático, Wu [17], em sua pesquisa, utiliza modelos formais como redes de Petri para especificação de atividades e objetos em um DDS. Em sua pesquisa, Wu [17], especifica atividades colaborativas, modelos de inteligência, ciclos de vidas de objetos em ambientes de DDS utilizando redes de Petri. Sobre uma mesma linha de raciocínio, Jalote [18], utiliza grafos para especificação formal de processos de rede. Para tanto, é necessário determinar conceitos básicos como Restrições operacional, restrição de conhecimento e restrição de recurso. Em seu projeto, Jalote [18], faz um escalonamento de tarefas baseado nas informações dos grafos.

3.1.4 Modelo de agrupamento

O conceito em que as unidades de desenvolvimento são abordadas sobre uma macro visão, constituindo uma organização única, é o agrupamento que não é apenas um recurso para a criação de organogramas, ao contrário, é um meio para coordenar o trabalho na organização. O

agrupamento, segundo Mintzberg [10] pode ter pelo menos três efeitos importantes: estabelece um sistema de supervisão comum entre as posições e as unidades, requerem posições e unidades para compartilhar recursos comuns, cria medidas de desempenho comuns e encoraja ao ajuste mútuo.

Há vários fatores que podem determinar o agrupamento de times. Mintzberg [10] denomina estes fatores como bases para o agrupamento. As bases nas quais as organizações agrupam posições e as unidades são indicadas na tabela 2.

Tabela 2 - Bases de agrupamento

Bases do agrupamento	Explicação
Conhecimento e habilidade	Constituído de acordo com a habilidade e o conhecimento das pessoas. Por exemplo, neste caso o agrupamento em DDS pode constituir-se em especialização por área tecnológica, como computação pervasiva, sistemas em tempo-real, etc.
Processo de trabalho e função	Composto por um sub-processo do todo ou uma função específica. Em um ambiente de DDS, pode significar, por exemplo, que o processo como um todo é todo subdividido em processos e os sub-processos são executados sequencialmente pelos <i>sites</i> .
Tempo	Tem o objetivo de aproveitar o tempo em diversos turnos de forma a maximizar a produtividade. Quando um software é desenvolvido ininterruptamente, em um ambiente de DDS (que neste caso é GSD) esta característica é denominada desenvolvimento <i>round-the-clock</i> .
Output	As unidades são formadas por produtos que fabricam ou por serviços que prestam. São agrupamentos, por exemplo, sobre linhas de produtos. Em um DDS, poderia identificar o tipo de produto que o <i>site</i> é especializado, por exemplo, sistemas ERP, sistemas CRM e tais como.
Cliente	Os grupos podem ser formados para lidar com diferentes tipos de clientes. Em um ambiente de DDS, neste caso, esta é uma possível abordagem em que as grandes empresas, não simplesmente montam um escritório próximo ao cliente, mas também todo o conjunto de desenvolvimento. As práticas de aproximar fisicamente a empresa ao cliente também é denominada de <i>nearshore</i> .
Local	Os grupos podem ser formados de acordo com o local na qual se encontram

3.1.5. Granularidade de repasse

Em um ambiente de desenvolvimento distribuído de software a granularidade do repasse pode ser definido de maneira diferente entre os *sites*. O repasse, por exemplo, pode ser de constituída de linhas de código, entretanto neste caso a codificação e as ferramentas são padronizadas entre os *sites*. Sobre esta abordagem de codificação, o *site* pode repassar o serviço para outro, independentemente do código fonte.

Segundo Jalote [18] a granularidade é definida principalmente pela homogeneização de processos, tecnologias e ferramentas da rede. A figura 3 é um exemplo de repasse, considerando a disciplina de codificação. Neste exemplo, a mais alta granularidade é repassar o serviço pronto para ser consumido (utilizado), e, a mais baixa granularidade é a linha de código, necessitando neste caso, ser acoplada a um método e/ou função.

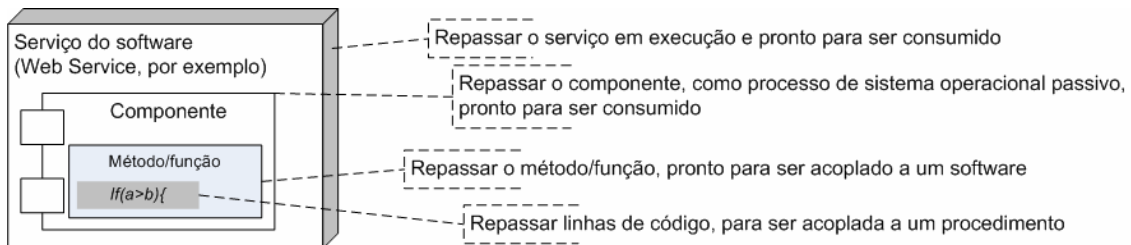


Figura 3 – Exemplo de granularidade do repasse na disciplina de codificação

4. APLICAÇÃO DO ESTUDO DE CASO

As propriedades indicadas neste trabalho foram obtidas a partir de dois projetos. Um deles, denominado ManWapp [19] e outro Tidia-EA [20]. Para tanto, as experiências relatadas no projeto ManWapp e as documentações do projeto Tidia, disponíveis em Incubadora [20], serviram de conteúdo para que estas propriedades fossem identificadas e validadas.

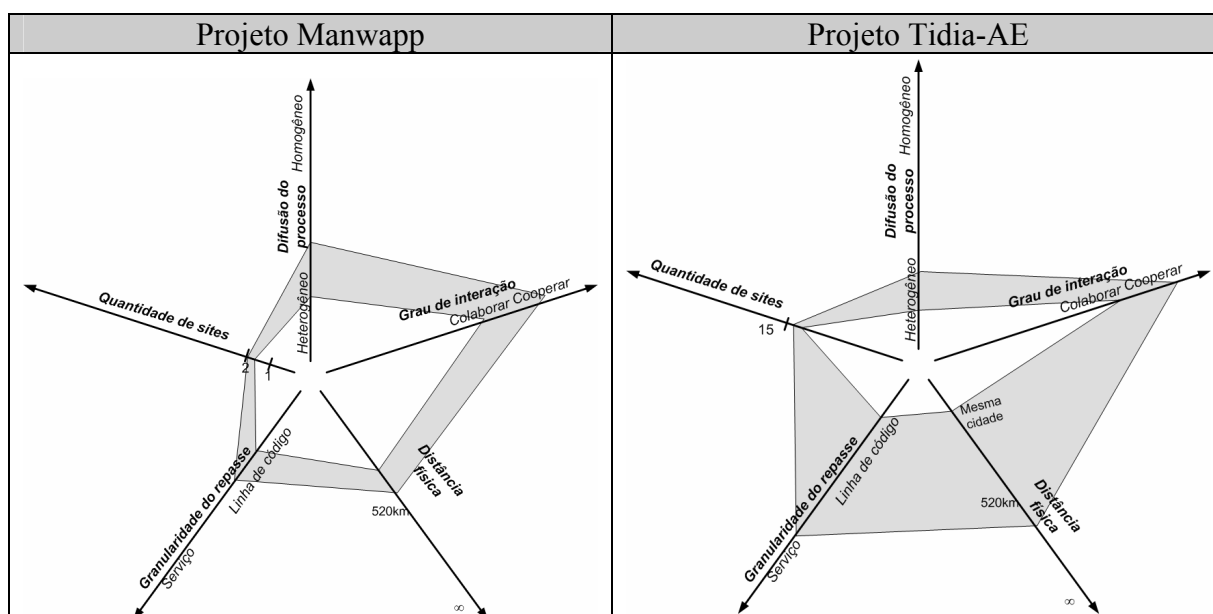
A metodologia empregada neste trabalho é o estudo de múltiplos casos. Este delineamento tem um caráter exploratório, uma plataforma para ajudar a intuição na recepção de intuições sobre a teorização de um dado problema. Um estudo de caso tem por principal objetivo ser a experiência prática que faz funcionar o modo de pensar empírico-indutivo, na construção de teorias. Esta pesquisa está alicerçada nos fundamentos de Yin [21].

O protocolo deste estudo está dividido em 3 seções. A primeira seção visa explorar a **organização da rede**, determinando seu perfil perante a teoria apresentada na seção 3 deste artigo. A segunda seção investiga como é o **gerenciamento de produção** dentro do ambiente da rede, identificando o distribuidor de tarefas e suas principais atividades. A última seção explora a **produção da rede**, revelando os efeitos das decisões do gerenciamento de produção.

As unidades de análise exploradas são os elementos envolvidos na pesquisa que o pesquisador aborda acordo com o seu ponto de vista. Nesta pesquisa as seguintes unidades de análise são abordadas: Interação entre nodos, desenvolvedores que trocam informações direta ou indiretamente com outro nodo e o processo de interação (sistemático ou *ad hoc*) entre os nodos.

Confrontando as variáveis indicadas na figura 2 conjuntamente com o resultado obtido após a aplicação do protocolo do estudo de caso, foram mapeados padrões indicados na tabela 3.

Tabela 3 – As propriedades de um ambiente de DDS em dois casos



Embora os casos sejam bastante distintos, pode-se perceber que ambos têm características comuns. Estas características incluem as vantagens, os desafios e também os problemas quando parte dos gráficos apresentados na tabela 3 são similares. Ainda neste estudo foram considerados os mecanismos de coordenação e as bases de agrupamento de cada caso.

Os pontos de similaridade estão indicados na figura 4 e uma comparação entre os projetos está indicada na tabela 4. Ambos os projetos foram desenvolvidos sobre um âmbito de cooperação, ou seja, desenvolvem software com um princípio de bem comum. Neste caso, os problemas foram similares nos dois projetos. A coordenação foi comprometida, uma vez que todos os *sites* já tinham um processo de software e em alguns casos utilizavam tecnologias diferentes. Inicialmente precisou ser estabelecido um conjunto de padrões iniciais, que em ambos os casos foram a tecnologia e os artefatos. As bases de agrupamento também foram similares. Em ambos os projetos, o projeto inicial foi subdividido e cada partição foi atribuída a um *site*. O impasse gerado, neste caso foi

grande, pois com a heterogeneidade dos nodos, a documentação ou até mesmo a compreensão dos segmentos do projeto tornaram-se ambíguos, gerando posteriormente impasses e retrabalho.

A difusão inicial de um processo de trabalho facilitou o desempenho da cooperação. Porém adequar processos em todos os nodos teve um custo de retrabalho. Um dos nodos do projeto Tidia – AE repassou codificação a uma outra organização. A documentação não foi compatível e conseqüentemente a organização refugou o trabalho. Neste caso houve retrabalho sobre a documentação a fim de adequá-la aos padrões compreensíveis a todos os nos.

O gerenciador de tarefas (papel responsável em alocar as tarefas na rede) precisou realocar trabalho e fazer os ajustes. Em ambos os casos o gerenciador de tarefas foi alocado sobre os *expertises* (papel responsável em coordenar e resolver problemas auxiliando o nodo a fim de seguir a padronização da rede) dos nodos. Cada nodo havia um *expertise*, que em uma reunião presencial discutiam como as tarefas seriam alocadas.

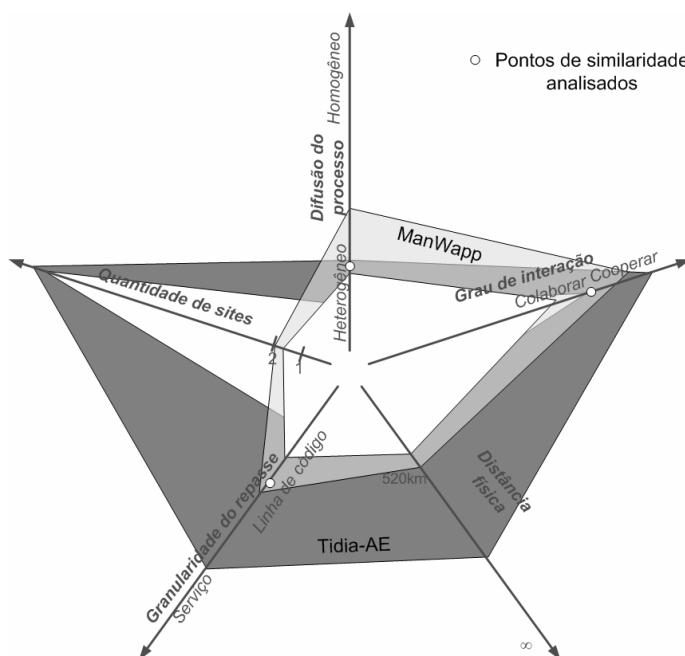


Figura 4 - Sobreposição dos projetos analisados

Tabela 4 - Comparação dos projetos analisados

Variável	Tidia - AE	ManWapp
Quantidade de sites	15	2
Distancia	450Km	De metros a muitos km
Grau de interação	Alto, sem histórico, (colaborativo)	Alto, sem histórico (colaborativo)
Difusão de processo	Padronização dos artefatos	Padronização dos artefatos e alguns processos de trabalho
Granularidade do repasse	Linhas de código, componentes, serviços e documentos	Componentes de software e linhas de código

O grau de interação foi intenso em ambos os projetos. Este fato ocorreu, pois ambos eram norteados pelo desenvolvimento cooperativo. Porém em uma exploração mais ampla, no projeto ManWapp, o histórico da comunicação entre os nodos foi pouco necessário. Este fato ocorreu pela quantidade mínima de nodos. Ao contrário disso, no projeto Tidia-ae, a ausência de histórico de comunicação afetou principalmente o ingresso de novos nodos na rede. Além disso, o histórico da comunicação pode ser empregado como mecanismo de ajuste mútuo entre os nodos operantes na rede.

No projeto ManWapp foram definidos um conjunto básico de produtos de trabalho que eram transitados na rede. A heterogeneidade do repasse presente no projeto Tidia-ae causou muitos impasses. A grande variedade de produtos de trabalho e também a variação da granularidade de repasse sobre um mesmo produto de trabalho, trouxe como conseqüência muitas recodificações. Estas recodificações ocorreram principalmente quando os componentes de software eram montados.

Em ambos os projetos, o desenvolvimento dos softwares, utilizaram soluções complexas, envolvendo arquitetura *multi-tier*, computação pervasiva, sincronização de recursos em sistemas distribuídos, software em tempo real e etc. Esta façanha foi possível graças à divisão do produto em vários módulos e tecnologias. Nestes projetos as especializações de cada *site* foram utilizadas. Cada *site* ficou responsável em desenvolver módulos que utilizavam tecnologias já conhecidas e difundidas no grupo. Com isto, o conhecimento foi disseminado entre os grupos por meio de *expertises*.

O volume de código desenvolvido em um curto período de tempo justificou o agrupamento em rede. Em ambos os projetos houve incentivo externo como os citados por Mintzberg [14]. Neste caso a consequência foi o agrupamento de um conjunto heterogêneo de *sites*. Os mecanismos de coordenação, junto com a difusão do processo tornaram-se evidentes como uma das principais variáveis. Este fato ocorreu, pois, a heterogeneidade dos nodos tanto na tecnologia quanto no processo de desenvolvimento precisaram de uma padronização inicial mínima para que os módulos de software desenvolvidos pudessem ser montados. Esta padronização envolveu padronização das interfaces do software, e de alguns processos de trabalho.

5. CONTRIBUIÇÕES

As contribuições deste trabalho almejam mensurar a eficiência de uma rede de produção de software. Para tanto, diversas teorias, como organização do trabalho [14], foram analisadas com o intuito de agregar valor à teoria de desenvolvimento distribuído de software. Desta maneira, foi possível estabelecer uma posição mais precisa da rede de produção, mapeando-a em variáveis que podem ser analisadas qualitativamente e/ou quantitativamente.

Diversos estudos de caso foram feitos na área, como é o caso de [5], [9] e [15]. Porém, não foi detectado em nenhum estudo de caso o posicionamento da rede. Para auxiliar uma análise sistemática de múltiplas redes de produção, algumas variáveis foram estudadas e analisadas inicialmente em dois estudos de casos ([19] e [20]). Este trabalho contribui na construção de um *framework* capaz de posicionar uma rede de produção de software. Com isso, é possível efetuar um estudo mais preciso, abordando diversas áreas multidisciplinares que compõem o desenvolvimento distribuído. Utilizando este *framework* inicial, embora os dois estudos de caso tivessem proporções distintas, as variáveis que se aproximaram em medida, refletiam em problemas semelhantes. Esta indicação inicial auxiliou na comparação dos dois casos, tanto quantitativamente quanto qualitativamente.

6. CONSIDERAÇÕES FINAIS

Este artigo apresentou um modelo de dinâmica para ambientes de desenvolvimento distribuído de software e suas propriedades levantados a partir de 2 estudos de caso. A configuração da rede, neste trabalho, inclui o conjunto que abrange o mecanismo de coordenação e as propriedades escalares, instanciados ao nível onde há efetivamente desenvolvimento que é o nível operacional da rede.

Nos projetos estudados, o desenvolvimento colaborativo foi possível, pois um mecanismo inicial de coordenação foi estipulado. Estes casos revelaram que neste ambiente onde os nodos eram demasiadamente heterogêneos, o ajuste mútuo como mecanismo de coordenação traz muitos impasses. Consequentemente, a padronização inicial de artefatos e de processos de trabalho, fizeram com que a produtividade aumentasse e a quantidade de impasses reduzisse.

Como trabalho futuro, um estudo sobre um ambiente de DDS onde os nodos são homogêneos, está em andamento. Neste caso há padronização dos produtos e processos de trabalho. O objetivo é elaborar uma comparação de ambientes homogêneos com heterogêneos.

A dinâmica do funcionamento e as propriedades discutidas neste trabalho, servem como arcabouço norteador para os gestores da rede. Por meio destes, é possível elaborar um mapeamento do estado atual da rede e, proativamente determinar os eventos de acordo com os atributos e os mecanismos de coordenação associados. Para os *stakeholders* que atuam na estratégia da rede, este mapeamento constitui a compreensão e posicionamento da rede como um todo, para os *sites*, representa o conjunto de relacionamentos entre eles e para os desenvolvedores, representa o processo em que estão engajados.

REFERENCIAS

- [1] ALSTYNE, Marshall Van. The state of network organization: a survey in three frameworks. in *Journal of Organizational Computing & Electronic Commerce*, 7(3); pp.83-151, 1997.
- [2] HERBSLEB, James D.; MOCKUS, Audris.. An Empirical Study of Speed and Communication in Globally-Distributed Software Development. **IEEE Transactions on Software Engineering**, v. 29, n. 3, p.1-14, 2003.
- [3] ANTUNES, P. **Groupware: Conceitos Fundamentais e Caracterização dos Principais Blocos Construtivos**. Relatório Técnico. Disponível em: <<http://www.di.fc.ul.pt/tech-reports>>. Acesso em: julho 2005.
- [4] GRUDIN, Jonathan. CSCW: History and Focus. **IEEE Computer**, New York, v. 27, n. 5, p.19-26, Maio 1994.
- [5] SUZUKI, Junichi; YAMAMOTO, Yoshikazu. Leveraging Distributed Software Development. **IEEE Computer**, New York, v. 32, n. 9, p.59-65, setembro 1999.
- [6] BECKER, Simon et al. A Delegation Based Model for Distributed Software Process Management. In: EUROPEAN WORKSHOP ON SOFTWARE PROCESS TECHNOLOGY, 8., 2001, Tóquio. **Proceedings...** Tóquio: ACM, 2001. p. 130 - 144.
- [7] MARTIN, Patrick. A Management Information Repository for Distributed Applications Management. In: INTERNATIONAL CONFERENCE ON PARALLEL AND DISTRIBUTED SYSTEMS, 2., 1996, Tóquio. **Conferência**. Tóquio: SBC, 1996.
- [8] SA, Jin; MASLOVA, Elena. A unified Process Support Framework for Global Software Development. In: INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE, 26., 2002, Oxford. **Proceedings...** Oxford: IEEE, 2002. p. 1 – 2.
- [9] CARMEL, Erran; AGARWAL, Ritu. Tactical Approaches for Alleviating Distance in Global Software Development. **IEEE Software**, v. 18, n. 2. março/abril 2001. p. 22-29.
- [10] MINTZBERG, Henry. **Criando organizações eficazes: Estruturas em cinco configurações**. 2. ed. São Paulo: Atlas, 2003. 334 p.
- [11] BORGHOFF, U., SCHLICHTER, J., *Computer-Supported Cooperative Work – Introduction to Distributed Applications*. 1. ed. Berlin: Springer, 2000. 529 p.

- [12] O'Day, V. L.; Bobrow, D. G.; Shirley, M. The social-technical design circle. Ackerman, M. S. ed. CSCW 96; Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work; 1996 November 16-20; Boston, MA. New York: ACM; 1996; 160-169.
- [13] BORGES, M. R., CAVALCANTI, M. C., MACHADO, M. L., Suporte por Computador ao Trabalho Cooperativo. XV Congresso da Sociedade Brasileira de Computação, Jornada de Atualização em Informática, Canela, 1995.
- [14] LOPES, L. T.; AUDY, J. L. N. Em busca de um modelo de referência para engenharia de requisitos em ambientes de desenvolvimento distribuído de software, WER03 - Workshop em Engenharia de Requisitos, Piracicaba – SP, Brasil, p329-342, 2003.
- [15] SHAMI, N. Sadat et al. An experimental simulation of multi-site software development. In: CONFERENCE OF THE CENTRE FOR ADVANCED STUDIES ON COLLABORATIVE RESEARCH, 10., 2004. **CASCON 2004**. 2004: ACM, 2004. p. 255 - 266.
- [16] MÜHLBACHER, Robert; NEUMANN, Gustaf. Towards a Framework for Collaborative Software Development of Business Application Systems. In: INTERNATIONAL WORKSHOPS ON ENABLING TECHNOLOGIES: INFRASTRUCTURES FOR COLLABORATIVE ENTERPRISES, 6., 1996, Stanford. **WET ICE '96**. Stanford: IEEE, 1996.
- [17] WU, Haihong; YU, Chunyan; WU, Minghui. Using Extended Fuzzy-Timing Petri Nets to Model Role-Based and Agent Oriented Collaborative Virtual Environment. In: WORLD CONGRESS ON INTELLIGENT CONTROL AND AUTOMATION, 5., 2004, Hangzhou. Hangzhou: IEEE, 2004.
- [18] JALOTE, Pankaj; JAIN, Gourav. Assigning Tasks in a 24-Hour Software Development Model. In: ASIA-PACIFIC SOFTWARE ENGINEERING CONFERENCE, 11., 2005, Busan. **APSEC'04**. Busan: IEEE, 2004.
- [19] L'ERARIO, Alexandre et al. Desenvolvimento distribuído de software para sistemas pervasivos: um estudo de caso. In: SIMPÓSIO BRASILEIRO DE SISTEMA DE INFORMAÇÃO, 1., 2004, Porto Alegre. **Simpósio**. Porto Alegre: SBC, 2004. p. 163 - 170.
- [20] INCUBADORA virtual da fapesp - TIDIA-EA. Disponível em: <<http://incubadora.fapesp.br/projects/tidia-ae/>>. Acesso em: 26 out. 2005.
- [21] YIN, Robert K.. **Estudo de caso: planejamento e métodos**. 3. ed. Porto Alegre: Bookman, 2005. 212 p.