1. Introdução

1.1. Motivação

A informática, que pode ser definida de uma forma geral como o processamento automático da informação, é utilizada hoje nos mais variados lugares. Dos robôs que ‘imitam’ humanos aos controles de fiados feitos por uma dona de uma barraquinha de salgados através de um computador de mão, quase todos os processos que podem ser automatizados estão sendo feitos através de algum sistema.

O uso de uma única tecnologia para automatizar a solução de um problema específico é bem comum. Porém, indo mais adiante, há casos em que é preciso realizar diversas tarefas, utilizando diferentes *software* para alcançar um único objetivo. É natural encontrar processos de negócio em empresas que envolvem duas ou mais etapas que são dependentes entre si, seja através de dados, quando uma utiliza dados provenientes da sua antecessora, ou de uma regra de negócio, quando é definido que uma etapa deve esperar pela execução da outra. As etapas podem ser consideradas como tarefas a serem realizadas dentro do processo. Neste cenário, há um fluxo de trabalho formado por estas tarefas, resultando em diferentes etapas, na tentativa de se alcançar um objetivo global. A este tipo de fluxo de trabalho dá-se o nome de *workflow*.

Os *workflows*, ou fluxos de processos, são comumente encontrados nas empresas, nas universidades e no dia a dia das pessoas. Diante disso, a sociedade acadêmica tem dado relevante atenção ao estudo dessa área procurando encontrar soluções interessantes para os problemas endereçados por esta tecnologia.

As ferramentas que permitem a definição e execução do *workflow* são chamadas de sistemas de gerência de *workflow* (*Workflow Management System* -

WfMS) (WfMC, 2005). Na arquitetura desses sistemas, as tarefas que compõem um *workflow* normalmente estão em diferentes máquinas (computadores) dispostas na Internet. Esta arquitetura distribuída é desejável, visto que ela possui diversas vantagens como distribuição de processamento, isolamento de problemas, reutilização de soluções já desenvolvidas, etc. Nesse ambiente, muitos dos sistemas de gerência de *workflow* são projetados assumindo uma conexão estável e confiável (Hackmann et al., 2006). Entretanto, existem casos que vão mais além dessa arquitetura básica, onde determinadas tarefas devem ser realizadas em campo, onde mobilidade é fundamental, e que não é possível a utilização desses tais tipos de máquinas, e, portanto, torna-se irrealista assumir que a conexão com a Internet é estável e confiável.

Como exemplos, podem ser citados funcionários de uma empresa que estejam visitando clientes: seja um representante de uma marca de refrigerante ou um vendedor de cosméticos; buscas de informações sobre plantas em uma floresta como parte da coleta de dados para uma pesquisa; processo de descoberta de focos de incêndio em uma área vasta e isolada como, por exemplo, uma mata; entre diversos outros exemplos que podem ser citados. Nestes casos, os dados associadas à tarefa são normalmente coletados de duas formas: manualmente, em que os dados são colhidos com anotações em papéis, ou através de um dispositivo eletrônico que já permita a inserção digital dos dados no local da realização da tarefa. Em um momento posterior os dados são levados para um ponto central que controla todo o processo e os dados são inseridos no *workflow*.

Observe que esses tipos de exemplos, em que parte do processo é realizada num determinado momento e a integração dos dados é feita posteriormente, são processos indesejáveis por não serem totalmente automatizados. Processos totalmente ou em parte manuais são inviáveis, pois as tarefas desenvolvidas manualmente podem afetar a eficiência do processo como um todo. Isto acontece porque a execução destas tarefas não é feita de forma automatizada.

Uma forma de automatizar esse tipo de *workflow* consiste em utilizar dispositivos móveis conectados à Internet como uma máquina participante do

processo. Com a utilização desses dispositivos, as tarefas antes manuais passam a integrar o *workflow* em tempo real. O uso de dispositivos móveis torna-se então uma solução interessante para a construção de um *workflow* com características de mobilidade o mais automatizado possível.

Dispositivos móveis ajudam a resolver o problema de tarefas antes realizadas de forma manual. Entretanto, eles trazem novos desafios. Um ambiente com suporte à desconexão envolve problemas como perda de mensagens, necessidade de sincronização de dados, etc., que não podem deixar de ser levados em consideração. A perda de mensagem pode impactar em todo o resultado de um *workflow*. Uma tarefa do *workflow* pode, por causa de uma eventual perda de mensagem não controlada, ficar esperando indefinidamente para iniciar sua execução.

1.2. Objetivos

Este trabalho tem como objetivo investigar questões relacionadas a sistemas de gerência de *workflows* em ambientes com suporte à desconexão utilizando dispositivos móveis.

Para alcançar tal objetivo será necessário criar mecanismos que possibilitem a distribuição das tarefas entre os dispositivos, gerenciar a troca de mensagens dentro do *workflow*, oferecer alternativas para tarefas que não podem esperar indefinidamente para começar sua execução e, principalmente, tratar o principal gargalo endereçado nesse tipo de ambiente: a desconexão.

Este trabalho apresentará uma arquitetura que cobrirá os objetivos do trabalho. A arquitetura proposta será basicamente a de um sistema de gerência de *workflow* (SGWf) com algumas extensões.

Como os dispositivos móveis participantes do processo possuem a característica de poder desconectar e conectar a qualquer momento, este trabalho contempla, além das questões comuns de SGWf: *triggers* de emergência, mecanismo sugerido para permitir que uma tarefa não fique indefinidamente esperando pelo início da sua execução; problemas endereçados pelo ambiente com desconexão, como, por exemplo, a perda de mensagem entre servidor e parceiros; a necessidade de reenviar a

tarefa cuja resposta não foi recebida, seja para o mesmo ou para outro parceiro; gerência do fluxo de informação de controle entre os parceiros, etc.

Este trabalho apresentará também uma discussão sobre conexões diretamente entre os dispositivos móveis.

Para validar a arquitetura proposta, este trabalho apresenta o MobiWfMS (*Mobile Workflow Management System*), um sistema de gerência de *workflow* com utilização de dispositivos móveis com suporte a desconexão.

1.3. Contribuições

A primeira contribuição desta dissertação consiste num resumo das principais tecnologias e teorias envolvidas com o trabalho. Serão abordados assuntos como: orquestração e coreografia de *workflow*; sistema de gerência de *workflow*, além das teorias envolvidas com *workflows* com desconexão.

A segunda, e mais importante, contribuição é uma arquitetura para sistemas de coordenação de *workflow* para ambientes com suporte à desconexão.

A terceira contribuição consiste na apresentação de um protótipo do sistema de coordenação de *workflow* para ambientes com desconexão desenvolvido para validar a arquitetura proposta.

1.4. Trabalhos relacionados

Esta seção resume alguns trabalhos que possuem tópicos relacionados a esta dissertação e que serviram de base para o trabalho pretendido.

MobiWork: Mobile Workflow for MANETs

MobiWork (Hackmann et al., 2006) é um sistema de gerência de *workflow* para MANETs (*Mobile Ad Hoc Network*). MANET é uma rede sem fio formada por dispositivos móveis, também sem fio, chamados de *hosts*. Nesta rede, os *hosts* se comunicam diretamente entre si e possuem características de desconexão.

A arquitetura do MobiWork contém componentes que dão suporte a operações de desconexão e mobilidade. Ela possui um algoritmo de alocação de tarefas entre os

participantes, nomeados de agentes. Este algoritmo designa tarefas para os dispositivos móveis, usando uma lista de restrições dinâmica associada com cada agente, baseada nas suas capacidades e habilidade de realizar as tarefas. São levados em consideração restrições como hora, local e duração de execução da tarefa, além da lista de qualificações exigidas para que se possa realizar a tarefa. Na alocação é verificado se o agente possui as características necessárias para realizar a tarefa e se ele está disponível, ou seja, não está executando outra tarefa, na hora e no local exigido para a realização da tarefa.

Essa dissertação não se preocupa com a alocação dinâmica de tarefas entre os participantes, mas, como poderá ser visto, apresentará uma alternativa para esta alocação dinâmica.

O protótipo do MobiWork foi desenvolvido em Java e assume o uso de *notebooks*, por serem mais robustos. A implementação não utiliza PDAs justificando que eles possuem diversas limitações de espaço e processamento como, por exemplo: as Java Virtual Machines para PDAs não são eficientes; a API gráfica do Java e a interface do usuário para estes dispositivos são restritas; as telas dos PDAs possuem baixa resolução. A comunicação entre os participantes é feita através de um *middleware* para comunicação *peer-to-peer*, o Limone (FOK et al., 2004), desenvolvido pelo mesmo grupo de pesquisa.

Essa dissertação admite, e tem o foco no uso de PDAs. No sentido contrário do MobiWork, apresentará uma arquitetura e um protótipo de sistema voltados para utilização em dispositivos móveis com baixo poder de processamento.

Disconnected operation in scientific workflow management systems

Este trabalho apresentado por (PAN, 200-) aborda questões de sistemas de gerência de *workflow* passíveis de desconexão. Os *workflows* científicos trabalham com grande volume de dados e com alto custo de processamento. Diante disso, uma desconexão ocorrida no sistema, seja por falha, ou até mesmo por desejo de um usuário, não pode fazer com que todo o trabalho realizado até momento seja perdido. Segundo ele, os problemas ocorrem, em sua maioria, por causa da centralização do

processamento num único coordenador do *workflow*, passando este a ser um ponto de falha.

Para tratar tal problema, são apresentadas quatro abordagens de arquitetura para os sistemas de gerência de *workflow*:

- **Abordagem descentralizada**, na qual o *workflow* é dividido em *sub-workflows* que serão coordenados por diferentes *hosts*;
- **Abordagem com *proxy***, em que os clientes com características de desconexão fazem uso de um *proxy* hospedado em um servidor com conexão mais estável. Neste caso, os clientes possuem uma maior autonomia de desconexão devido ao fato do trabalho está sendo coordenado através do *proxy*;
- **Abordagem distribuída**: não existe a figura do controlador central, o *workflow* é coordenado por todos os participantes do processo. Cada participante, apesar de realizar apenas uma parte do *workflow*, tem total conhecimento do mesmo, sabendo assim onde buscar informações quando necessário;
- **Abordagem híbrida**: é uma junção da abordagem com *proxy* e da abordagem distribuída. Faz uso das vantagens, ao mesmo tempo em que tenta suprir as deficiências apresentadas pelas duas abordagens.

Este trabalho contribui com essa dissertação ao apresentar os problemas de desconexão, que seria ocasionado pela centralização do processamento e ao sugerir abordagens de arquitetura que servirão de embasamento para a construção da nossa arquitetura.

Exótica/FMDC: Handling disconnected clients in a workflow management system

Em Exótica/FMDC (Alonso et al., 1995) é apresentado um modelo de sistema de gerência de *workflow* com característica de desconexão. As técnicas utilizadas para o controle de desconexão são implementadas baseada na arquitetura do WorkFarm, um WfMS da IBM projetado para gerência de processos de negócio. Neste sistema há um servidor central que controla todo o processo, fazendo uso de uma base de dados

centralizada que contém toda a informação utilizada durante o processo. Uma das atividades do servidor é gerenciar, dentre os clientes aptos para realizar uma determinada atividade, quem vai executá-la. Este controle é feito através de trocas de mensagens, assumindo de início que há uma conexão permanente e confiável entre os clientes e o servidor.

Para adicionar suporte à desconexão, o Exótica/FMDC propõe que se dê mais autonomia para os clientes. Para isso, sugere que os componentes instalados na máquina cliente passem a ter sua própria base de dados com dados suficientes para que cada atividade seja executada. Assim serão capazes de continuar com o processamento sem precisar consultar a base de dados central a cada passo.

O servidor deve garantir que não haja conflito entre dois clientes desconectados, de forma que eles não executem a mesma tarefa simultaneamente. Para tratar isso é feita uma lista de tarefas bloqueadas. Os clientes devem notificar suas intenções de realizar uma tarefa quando desconectados, para que o servidor possa evitar conflitos. Cada vez que um cliente expressa o desejo de realizar a atividade, a mesma é bloqueada de forma que outros clientes não conseguem executá-la também. Isto pode ser informado pelo cliente no primeiro momento em que ele estiver carregando os dados do servidor.

No momento de re-conexão, etapa considerada pelo Exótica/FMDC, os clientes e o servidor trocam as mensagens para a sincronização dos dados.

O Exótica/FMDC contribui com essa dissertação através dos mecanismos de desconexão utilizados que servirão de inspiração para os mecanismos adotados nessa dissertação. O Exótica/FMDC, diferentemente dessa dissertação, não admite o uso de máquinas clientes com baixo poder de processamento e mobilidade.

Esta dissertação utiliza os trabalhos relacionados como base para apresentar uma arquitetura coesa com suporte a dispositivos móveis com características de desconexão, gerenciamento de falhas e suporte a atividades de emergência.

1.5. Organização da dissertação

Além desse capítulo, que apresenta a introdução deste trabalho, esta dissertação está organizada da seguinte forma. O capítulo 2 faz um levantamento dos conceitos básicos relacionados ao tema deste trabalho. O capítulo 3 mostra os requisitos e as restrições necessários para o trabalho. O capítulo 4 descreve a linguagem de *workflow* definida neste trabalho. O capítulo 5 apresenta a arquitetura proposta para sistema de gerência de *workflow* em ambientes com suporte a dispositivos móveis. O capítulo 6 mostra uma implementação desenvolvida para validar a arquitetura proposta. Por fim, o capítulo 7 conclui o trabalho e propõe alguns trabalhos futuros.