

Propostas para Trabalhos de Mestrado no âmbito do Projeto Xchart

- Linguagem de Configuração
Uma vez descrito em Xchart, como iniciar a execução de um sistema assim descrito? A proposta é prover um mecanismo para (a) especificar uma configuração de um sistema desta natureza e o seu "estado inicial", bem como (b) compilar a especificação de tal configuração e do seu estado inicial para produzir um script capaz de "levantar" o sistema conforme especificado e deixá-lo pronto para uso.
- Sistema de Execução (núcleo reativo)
Exploração de formas alternativas de implementação do Núcleo Reativo (NR) com ênfase inicial em facilidade de modificação para permitir experimentações diversas em nível semântico. Percurso eficiente na árvore de um Xchart e implementação eficiente de tabelas de decisão são alguns dos tópicos envolvidos.
- Framework Xchart (arquitetura de software)
Como os componentes do ambiente Xchart poderão estar integrados e cooperar entre si? O objetivo é a definição de um "esboço" de um "roadmap" para futuros trabalhos: caracterizar componentes e suas interfaces, possibilidades de composição de tais componentes, ...
- Ambiente de Teste
Como testar Xchart? Quais propriedades a serem verificadas nos casos de teste? Há alguma análise de diagramas que possa ser relevante? A especificação completa, isto é, o controle definido pelo diagrama mais a semântica definida por atividades, precisa ser considerada? Para verificar a preservação de determinadas propriedades, talvez seja necessário usar diversas instâncias de um mesmo Xchart. Seria possível traçar um paralelo com o framework JUnit utilizado no contexto de geração de código?
- Simulador
Definição e implementação de um simulador. Embora existam "similaridades" entre as funcionalidades de um editor e um simulador de especificações Xchart, há considerações específicas e outros desafios. Um bom framework poderia facilitar o uso comum de código entre o simulador e o editor. O simulador é diferente do ambiente de teste, pois ele levará em conta apenas as definições de controle, isto é, os diagramas Xchart. É preciso definir uma "scripting language" em que seja possível dizer quantas instâncias de cada diagrama pertinente sejam necessárias para a sessão de simulação e a seqüência de eventos (locais e globais) a serem gerados. É possível imaginar dois cenários: no primeiro, o disparo do próximo evento da lista só irá ocorrer depois que o "sistema" (sem a parte semântica) estiver estabilizado novamente e, no segundo, define-se exatamente em que momento tal evento será disparado. O segundo cenário será útil para simular situações de stress. A lista de eventos, ao invés de uma simples lista de eventos, poderia conter alguns "comandos" que indicassem repetições de disparos de um ou mais eventos e "condições de guarda". Além da linguagem é preciso criar uma ferramenta que interpreta tais scripts e "simula" a execução do "sistema" (sem a parte semântica) com o auxílio do sistema de execução.
- Integração de Xchart com frameworks que implementam o padrão MVC
Atualmente existem vários frameworks que implementam o padrão MVC. Spring MVC e Struts são dois exemplos. Este trabalho visa investigar e propor uma forma de integração de Xchart com um framework dessa natureza. Facilidade de integração de Xchart com estes frameworks, sem "costuras", é o principal resultado esperado deste trabalho.
- Integração de Xchart com Workflow
Especificações em Xchart, por serem potencialmente distribuíveis, podem desempenhar o papel de elementos de ligação entre diferentes engenhos de workflow executando em servidores distintos. No âmbito do trabalho proposto cabe investigar se a linguagem Xchart é capaz de descrever, em parte ou no todo, processos de trabalho (workflows) e desenvolver

um compilador que gere especificações textuais de workflows de acordo com os padrões pertinentes.

- Compilador Xchart

Há duas “linguagens” imprescindíveis para a execução de Xchart: uma linguagem de especificação de Xchart através da qual diagramas nesta linguagem gráfica são registrados de forma passível de serem processados por um programa e uma linguagem adequada para a execução do Sistema de Execução de Xchart.

A execução de um Xchart depende da existência da especificação correspondente. Esta especificação deverá ser produzida em conformidade com uma linguagem de especificação, provavelmente “oculta” do usuário de Xchart pelo emprego de um editor (para este usuário um Xchart é um diagrama). Esta linguagem tem o propósito de registrar especificações em Xchart. É através do emprego desta linguagem que um usuário poderá confeccionar especificações através de várias sessões sem que a posição de estados e aspectos visuais, por exemplo, não sejam perdidos entre uma sessão e outra.

Ao mesmo tempo em que a posição relativa de estados é relevante para o registro de uma especificação, não o é para o Sistema de Execução de Xchart. Por outro lado, condições e ações podem ser compiladas em bytecodes para evitar uma interpretação das operações correspondentes e, dessa forma, aumentar o desempenho da execução de um Xchart. Esta geração dinâmica de código pode ser utilizada por outros módulos como o simulador de Xchart.

Embora o Sistema de Execução de Xchart possa desconsiderar o que não é relevante para a execução e realizar, em tempo de execução, transformações úteis à execução, tais operações podem ser realizadas por um compilador. Além de permitir uma melhor divisão de tarefas entre componentes, viabilizaria o uso do Sistema de Execução em dispositivos móveis com poucos recursos para as exigências de um compilador.

A definição da linguagem de especificação de Xchart e da linguagem de saída, produzida pelo compilador, o compilador e a definição de requisitos desejáveis nestes itens são temas deste trabalho.

- IDE Xchart (edição, depuração, ...)

A implementação de partes do IDE está contemplada em várias propostas identificadas acima. Este item aqui refere-se apenas à definição (requisitos) do IDE. O Eclipse (<http://www.eclipse.org/>) pode servir como plataforma base para tal ambiente. É preciso projetar algo de fato útil, fácil de ser empregado no cotidiano. A preferência pelo Eclipse se deve ao fato de ser um ambiente que se tornou um padrão de fato e da sua arquitetura permitir a adição de extensões. Com um ambiente deste tipo é possível produzir algo de mais fácil uso pois é possível ter uma integração mais forte de ferramentas que permitirão ao desenvolvedor atuar em níveis de abstração distintos sem causar discontinuidades.