
CAPÍTULO 10

CONCLUSÕES E TRABALHO FUTURO

ÍNDICE

10.1 - Sinopse e Contributos da Tese.	299
10.2 - Conclusões e Trabalho Futuro.	303

10.1.- Sinopse e Contributos da Tese.

A figura 10.1 ilustra a filosofia do desenvolvimento de sistemas, segundo [Ghezzi et al. 91], que irá servir de referencial na apresentação da sinopse da tese e seus contributos.

Cada filosofia de desenvolvimento de sistemas, ou *paradigma*, assenta num conjunto de métodos que corporizam um conjunto de regras visando a sistema-tização dos processos. Estabelecidos os métodos, surgem de seguida todos os auxiliares tecnológicos, que não só os materializam e provam a sua viabilidade prática, como servem de instrumentos à sua efectiva aplicação. Naturalmente que, no limite mais exterior, se encontram as imprescindíveis notações, que são a garantia de possibilidade de registo e de comunicação.

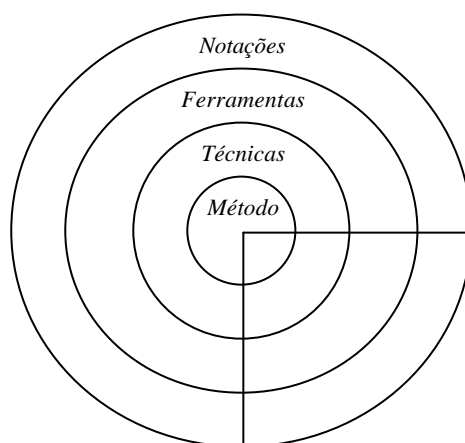


Fig. 10.1 - Filosofia de Desenvolvimento de Sistemas.

O trabalho realizado nesta tese, não sendo a proposta completa e coerente de um novo paradigma de interacção, é, ainda assim, um esforço metodológico no sentido da concepção sistemática e rigorosa de Sistemas Interactivos. É deste modo natural que se lhe reconheçam componentes semelhantes às que acima se consideram constituir uma verdadeira filosofia de desenvolvimento de sistemas: *Método, Modelos e Técnicas, Ferramentas e Notações*.

Cada um dos contributos da tese será de seguida apresentado e discutido em função do seu posicionamento em tal conjunto de componentes.

Método.

O desenvolvimento de Sistemas Interactivos tem sido principalmente realizado seguindo as regras e facilidades oferecidas pela inúmera tecnologia disponível para a construção da IU e, conseqüentemente, sem grande apoio metodológico ou de sistematização.

Em resultado, o projecto de Sistemas Interactivos é hoje, em larga medida, realizado sem que haja uma clara ligação entre a construção da camada compu-tacional e a construção da camada interactiva. Na maioria das vezes tal ligação é estabelecida já numa fase muito avançada do projecto, ligando o código final da aplicação, em geral de modo “ad hoc”, a objectos interactivos disponíveis na tecnologia de interacção que se adopta.

A possibilidade de se encontrar um método de desenvolvimento de base ri-gorosa, sistematizador do processo e facilitando a integração das concepções de cada uma das camadas desde a fase inicial do projecto, constituiu a principal motivação para o trabalho desenvolvido. Como premissa fundamental, ou ponto de partida, assumiu-se que para a concepção e desenvolvimento da camada computacional, métodos formais de especificação baseados em modelos mate-máticos seriam utilizados, não só pelo seu rigor notacional mas também porque, em geral, possibilitam a construção de modelos executáveis das aplicações, i.é., protótipos. No entanto, procurou-se que o não emprego de métodos formais para a camada computacional não fosse impeditivo da aplicação do método.

O método de especificação adoptado, CAMILA/SETS, tem um poder expressivo equivalente a outros métodos mais divulgados de especificação baseados em modelos, tais como VDM e Z, apresentando ainda vantagens adicionais. Por um lado permite a construção de protótipos funcionais, ou até baseados em proces-sos, o que não é comum. Por outro lado, dado ter a si associado um cálculo ri-goroso (SETS), possibilita o refinamento de especificações por cálculo e não pelo mais complexo mecanismo de refinamento sucessivo empregue, por exemplo, em VDM ou Z.

Fixado um método para o desenvolvimento da camada computacional baseado em especificações formais, prototipagem rápida e iteração sobre o protóti-po dos requisitos funcionais definidos para a aplicação, o método integrador a encontrar deveria, antes de mais, ser compatível com este. A independência da concepção da camada computacional é garantida à partida, restando agora ga-rantir o mesmo grau de independência para a concepção da camada interactiva, concepção essa a ser realizada já sobre informação sintáctica e semântica rela-tiva à aplicação, e em coerência com esta.

A utilização de um ciclo de concepção da camada interactiva semelhante ao empregue para a camada computacional, ou seja, um ciclo baseado em prototi-pagem e iteração, tornou-se uma ideia atractiva, tanto mais que a prática vigen-te na área, ainda que em condições diferentes¹, é maioritariamente iterativa (cf. 87% dos casos encontrados segundo [Meyers 92]).

¹ Em geral, é realizada iteração sobre aspectos de apresentação da IU, dado não existir informação semântica suficiente sobre a aplicação, sendo no entanto esta informação determinante do “feel” e de parte do “look” da IU.

O método a definir para a concepção e desenvolvimento da IU deveria, por-tanto, não só garantir que a informação semântica necessária à construção da camada interactiva pudesse ser disponibilizada a partir da especificação da camada computacional, como também que tal informação servisse de base à criação de protótipos da IU visando concepção iterativa.

Foi assim tomada a decisão de que o protótipo da IU gerado deveria possuir de forma concreta ou simbólica, toda a informação semântica necessária para a correcta reprodução do comportamento final a apresentar pela IU. Sendo a camada computacional, nesta fase inicial do projecto, apenas uma especificação formal animada pelo respectivo protótipo, então o método a seguir na construção do protótipo da IU deveria basear-se apenas em tal informação sobre a camada computacional.

O método proposto baseia-se assim na inferência sistemática de informação relevante para a construção da IU a partir de informação disponibilizada na especificação por modelos da camada computacional, em particular, na informação da funcionalidade das operações, nas respectivas pré-condições e nos invariantes de dados. Esta informação, de índole matemática, revelou-se suficiente-mente rigorosa e completa para que protótipos da IU pudessem ser de imediato criados, recorrendo a ferramentas auxiliares. Estes protótipos da IU exibem des-de logo um comportamento (“feel”) semelhante ao esperado para a IU final, ainda que sejam executados em ligação ao protótipo da camada computacional.

A iteração sobre tal protótipo da IU, caso não haja alteração de requisitos na camada computacional, visa sobretudo ajustar e completar a sua apresentação. Este resultado é, só por si, de bastante interesse, em particular se recordarmos que a maioria das abordagens referidas apenas permite a prototipagem da IU ou em ligação à aplicação final ou em isolamento, ou seja, sem contexto semântico. Na abordagem apresentada, a ligação à camada computacional está já estabelecida, através de um modelo desta, podendo as duas concepções prosseguir em separado seguindo técnicas próprias. A camada computacional poderá ser agora refinada segundo diferentes métodos, e a camada interactiva, naquilo que pode ser ajustado, ser construída por exemplo sob directivas de especialistas em Ergonomia ou Factores Humanos. Conseguiu-se assim separação suficiente para que, quanto aos aspectos específicos e independentes, tais concepções possam evoluir em isolamento, provendo no entanto o protótipo da IU com a informação contextual crucial para o seu desenvolvimento.

Modelos e Técnicas.

A possibilidade de se possuir contexto semântico, ainda que modelado, numa fase muito inicial do projecto, levou à definição de um modelo de interacção capaz de tirar partido de tal informação, prestando ao utilizador assistência sin-táctica e, principalmente, semântica. As interfaces construídas segundo

tal mo-delo exibem um comportamento caracterizado pela impossibilidade de ocorrência de erros, quer sintácticos quer semânticos.

O objectivo de se garantir interacção sintacticamente correcta, conduziu à definição de um modelo de interacção baseado em *arquétipos* e em edição/cons-trução incremental de objectos guiada pela sua representação estrutural. Os *arquétipos* foram formalizados como sendo *termos com variáveis* gerados a partir de uma álgebra de manufactura dos objectos em edição. Este modelo de arqué-tipos, em conjunto com as operações para a sua construção e manipulação que se apresentaram, possui características de representação e manipulação simbó-lica a explorar em sistemas onde representações explícitas de objectos ou do seu processo de construção devam ser consideradas².

A necessidade de introdução de assistência semântica em conjunto com a assistência sintáctica, conduziu à definição de um modelo de interacção gené-rico designado *Modo Assistido Sintáctico-Semântico* (MASS), impondo um conjun-to de requisitos às IU desenvolvidas em obediência ao mesmo. O objectivo prin-cipal é, naturalmente, aumentar o grau de facilidade de utilização das IU desen-volvidas, incluindo preocupações de aumento de "usabilidade" a par das preocu-pações de rigor, sistematização e integração. A obediência ao modelo proposto permitiu a criação de IU possuindo um comportamento "profiláctico", ou seja, preventivo relativamente à geração de erros sintácticos ou semânticos.

O fornecimento ao utilizador de informação semântica adequada realiza-se segundo um mecanismo designado por "feedforward", que corresponde ao cálcu-lo do conjunto de valores que, no actual contexto de interacção, podem ser con-siderados e aceites como válidos.

Em resultado de tal conjunto de decisões, e ainda que permitindo que para cada IU utilizável em modo MASS um modo não-MASS, baseado numa lingua-gem de comandos, pudesse ser igualmente usado, a camada computacional po-de ser em geral vista como um conjunto de funções totais, ou seja, sempre cor-rectamente invocadas, sintáctica e semanticamente.

Ferramentas.

A necessidade de materialização destas ideias e destes modelos, conduziu à cri-ação de um ambiente para a geração semi-automática de sistemas interactivos possuindo IU seguindo o modelo MASS, o sistema GAMA-X, tendo-se igualmente explorado a via da implementação do MASS em aplicações já existentes, visando a sua recuperação tecnológica do ponto de vista interactivo.

² Um exemplo actualmente em desenvolvimento permite a especificação e a construção de "plantas de habitações", cuja semântica pictórica se transporta para AutoCAD? via AutoLisp? [Oliveira 95].

No entanto, e para aplicações a construir de raiz, o sistema GAMA-X é a ar-quitectura que se propõe para apoio à sua geração semi-automática. A partir de especificações formais da aplicação, e seguindo o método sistemático, são pro-duzidas especificações do comportamento do controlador do diálogo do sistema, e dos modelos da aplicação e da apresentação. A partir destas especificações é produzida a IU, de imediato ligada ao protótipo da aplicação e pronta para que a sua apresentação final possa ser ajustada por iteração.

A escrita e introdução das especificações do controlador do diálogo necessá-rias a geração automática deste pelo GAMA-X, é suportada por um editor estru-turado que reconhece a correcção sintáctica de tais especificações e gera até, de modo automático, o código de comportamento do controlador.

Arquitecturas são plataformas reais para a descrição de implementações. Para além deste mérito, muita da sua importância reside também no facto de que, perante uma arquitectura concreta, é necessário que os princípios, regras e técnicas anteriormente divisadas em abstracto, possuam uma tradução a este nível concreto. Só assim haverá a garantia da sua viabilidade tecnológica. O sis-tema GAMA-X, para além de ser um gerador de IU, é o sistema que garante a via-bilidade das ideias apresentadas.

Ainda no campo das ferramentas, foi também desenvolvido um sistema para a geração de interfaces adaptativas, o sistema GAIA, de momento um protótipo, e que se pretende que venha a incorporar a arquitectura do GAMA-X, ainda que as suas características lhe permitam um funcionamento autónomo.

Notações.

Finalmente, e como a figura 10.1 ilustra, para que todos os conhecimentos, de-cisões e directivas se comuniquem, necessitamos de notações. As principais di-rectivas são, neste caso, as que devem ser comunicadas ao GAMA-X para que es-te possa gerar automaticamente o controlador do diálogo.

Os *Guiões de Interação* são o formalismo criado para a escrita de tais espe-cificações. Conceptualmente construídos à volta da noção de *síntese de um co-mando*, apresentando assim um primeiro nível de modularidade conceptual que facilita a sua utilização, os guiões oferecem ainda composicionalidade, permi-tindo construir especificações de forma modular e estruturada. O seu poder ex-pressivo foi demonstrado em diversos exemplos, tendo aí sido comparado com notações de especificação baseadas em processos.

A semântica operacional dos guiões foi definida à custa do formalismo das Redes de Petri Etiquetadas-Guardadas-Interruptíveis, sendo estas redes

geradas automaticamente a partir das especificações dos guiões. O tratamento de alguns eventos assíncronos foi também incorporado.

10.2.- Conclusões e Trabalho Futuro.

Os elevados custos hoje ainda imputáveis ao desenvolvimento de IU, apesar da tecnologia disponível, só podem ser diminuídos se no processo de desenvolvimento se introduzir sistematização e geração, pelo menos, parcialmente auto-matizada. O autor não crê na automatização total do processo, mas acredita que a tendência seja para, quer para a camada interactiva, quer para a computacional, cada vez seja possível gerar automaticamente mais componentes, até um dado limite.

No caso da IU, a “apresentação” final e, eventualmente, aspectos flexíveis de comportamento, deverão ser sempre ajustáveis por iteração, pelo utilizador ou junto deste. A especificação da apresentação é uma tarefa complexa, porque de muito baixo nível, e, pelas razões expostas, provavelmente de custos demasiado elevados relativamente à garantia de eficácia final.

Um aspecto em geral negligenciado no desenvolvimento de IU é o tratamento do “output”. No sistema GAMA-X o “output” foi tipificado, tendo-se construído um conjunto de objectos de interacção específicos para certas estruturas, sendo tal interacção vista como atómica. A mesma abordagem deverá ser realizada para o tratamento dos objectos resultantes de informação oriunda da aplicação, que deverá poder ser visualizada e imediatamente utilizável pelo utilizador.

Ainda que de forma simples de momento, o GAMA-X vai já de encontro a necessidades que se começam a sentir, no sentido de que os SGIU deixem de ser orientados ao evento e passem a ser orientados à (cada vez mais volumosa) informação e à comunicação com APIs.

Um aspecto preocupante das IU é o seu carácter efémero apesar dos custos elevados do seu desenvolvimento. Note-se que a camada computacional, principalmente se “refugiada” sob o escudo de uma API, irá sempre sobreviver. A IU, à medida que certas operações manuais vão sendo automatizadas, passa a perder funcionalidade, entretanto substituída pela de uma outra IU. No limite, a primeira IU poderá desaparecer dando lugar a uma nova.

Por outro lado, assiste-se hoje ao aparecimento de tecnologia de suporte que permite que aplicações e respectivas IU, até então isoladas porque incomunicáveis, passem rapidamente a poder intercombinar-se pela intercombinação das suas IU (ou parte delas)³.

³ Veja-se o que se passa com aplicações em ambiente Windows? com o aparecimento da tecnologia OLE (“Object Linking and Embedding”) [Clark 92].

O trabalho a realizar no sistema GAMA-X, para além da necessidade de se colmatarem as insuficiências já apontadas, prosseguirá no sentido de o prover de capacidades que possam acompanhar os mais recentes desenvolvimentos tecnológicos, aumentando a sua capacidade de exprimir interacção (quer quanto a dados quer quanto a modos).

É pois previsível, em resultado de tais avanços tecnológicos, que novos problemas de falta de rigor e de metodologia surjam, o que certamente conduzirá à procura de novos formalismos e métodos em Ciências da Computação. Torna-se no entanto importante salientar, em resultado até da experiência neste trabalho, que a multidisciplinaridade, removida a complexidade, pode ser vista como uma cooperação mutuamente frutuosa. Formalidade sem contexto será sempre pouco útil.