



Escola de Engenharia
Departamento de Informática
Universidade do Minho

Lígia Iolanda Macedo Chiuale

Especificação do modelo de folha
de cálculo avançado

Maio 2018

Universidade do Minho

Escola de Engenharia

Lígia Iolanda Macedo Chiuale

Especificação do modelo de folha de Cálculo avançado

Tese de Mestrado

Mestrado em Engenharia Informática

Trabalho realizado sob a orientação de:

Doutor João Saraiva

Anexo 3
Declaração

Agradecimento

Em primeiro lugar, gostaria de agradecer ao meu supervisor e co-supervisor o Prof. Dr. João Saraiva e o Mestre Jorge Mendes, respectivamente, por me concederem a oportunidade de trabalhar em tal assunto interessante, e pelo enorme apoio que me foi dado durante a duração desta tese.

Por último, gostaria de dar um agradecimento muito especial às pessoas que me encorajaram em nos últimos dois anos, e minha família, especialmente meus filhos Ladislau Atílio, Daliana Selene, a minha mãe, Elvira Macedo, meu tio Joaquim Macedo, os meus irmãos, os meus colegas e amigos Pedro Paulo e Dino, a minha querida prima Carla Macedo. Também gostaria de agradecer a Deus pela sabedoria e paciência que me deu, sem ele nada só.

Resumo

Especificação do modelo de folha de Cálculo avançado

As folhas de cálculo são muito usadas por programadores e não programadores. Mas, foi revelado, mediante estudos que muitas folhas de cálculo construídas contêm erros, que causam impactos negativos gestão do negócio. Para melhorar o processo de desenvolvimento da folha de cálculo foi introduzido técnicas baseadas na abordagem orientado por modelos, tal como templates de folha de cálculo, ClassSheets e diagrama de classes. Entretanto, pretendemos com este estudo analisar a expressividade das extensões do modelo de folha de cálculo proposto por Mendes na sua tese de Doutoramento.

Para realizar este trabalho, revisamos os ClassSheets e as extensões do modelo de ClassSheets. Depois, observamos os tipos de estruturas de folhas de cálculo usadas no contexto real. Com base os padrões observados nas folhas de cálculo, avaliamos a expressividade dos modelos ClassSheets. Adicionalmente, na prática, apresentamos a especificação do modelo folha de cálculo usando os ClassSheet e verificamos insuficiência da linguagem de ClassSheets. Com base a especificação do modelo de folhas de cálculo usada num contexto real, foi proposto um novo mecanismo, chamado, botão recursivo. Usando este mecanismo o utilizador pode criar modelos de folha de cálculo com item composto, organizados com indentação. No final, avaliamos o botão recursivo usando um exemplo de folha de cálculo real e representamos o modelode folha de cálculo avançado e geramos a respectiva instância. Os resultados deste trabalho mostraram que ClassSheets tradicionais permitem apenas representar folhas de cálculo com estruturas de tabelas simples.

Abstract

Advanced specification spreadsheet models

Spreadsheets are widely used by programmers and non-programmers. But, many studies have shown that spreadsheets contain errors. Many of these errors have significant impacts on business management. To improve this process of spreadsheet development, we introduced techniques based on the model-driven approach, such as spreadsheet templates, ClassSheets and class diagrams. With this study, we intend to analyze the expressiveness of the ClassSheets model extension that was proposed by Jorge Mendes in his doctoral thesis.

To perform this work, we review the ClassSheets and extensions of the ClassSheets models; we look at the types of spreadsheet structures used in the industrial context. Based on the patterns found in spreadsheets, we evaluated the expressiveness of the ClassSheets models and their inadequacies. Next, we present the spreadsheet specification using the ClassSheets models and verify the insufficiency of the ClassSheet language. Based on the spreadsheet specification used in the industrial context, a recursive mechanism was proposed, called recursive button. Using this mechanism, it is possible for the user to specify a spreadsheet with compound items. Finally, we evaluated the new mechanism using a real spreadsheet example. The results showed that ClassSheets allows representing spreadsheets with structures of simple tables.

Índice de conteúdos

1. Introdução	1
1.1 Folhas de cálculo	1
1.2 Motivação	2
1.3 Objectivo	4
1.4 Estrutura da dissertação	4
2. Folhas de cálculo dirigidas por modelos	5
2.1 Templates de folha de cálculo	5
2.2 ClassSheets	7
2.3 Especificação do modelo da folha de cálculo usando o ClassSheets	10
2.4 Resumo	12
3. Modelo de folha de cálculo Avançado	13
3.1 Mecanismo recursivo	13
3.2 Validação do botão recursivo	14
3.3 Modelo de folha de cálculo avançado	19
3.4 Resumo	21
4. Conclusão	22
5. Bibliografia	23

Índices de figuras

FIG 1. 1: PLIMPTON 322 – UMA TABLETE DATADA DE VOLTA DE 1800 BC.....	1
FIG 1. 2: FOLHA DE CÁLCULO EM PAPEL.....	2
FIG 2. 1: TEMPLATE DA FOLHA DE CÁLCULO PILOTOS.....	6
FIG 2. 2: SINTAXE DO TEMPLATES VITSL.....	7
FIG 2. 3: SINTAXE DA CLASSHEET. FONTE[1].....	8
FIG 2. 4: EXEMPLO PILOTS.....	9
FIG 2. 5: INSTÂNCIA DA TABELA DE ANÁLISE FINANCEIRA DE CUSTOS, DA FOLHA DE CÁLCULO ADAPTADA, FICHEIRO 'FINANCIAL/PROCESSED/COSTFACTORS.XLS', DO CORPUS DE FOLHAS DE CÁLCULO EUSES [17].	10
FIG 2. 6: MODELO DE CLASSHEET ANÁLISE FINANCEIRA DE CUSTOS.....	11
FIG 3. 1: BOTÃO RECURSIO.....	13
FIG 3. 2: MODELO DE FOLHA DE CÁLCULO AVANÇADO.....	15
FIG 3. 3 : 1º INSTÂNCIA DO MODELO	16
FIG 3. 4 : 2º INSTÂNCIA DO MODELO	16
FIG 3. 5 : 3º INSTÂNCIA DO MODELO	17
FIG 3. 6 : 4º INSTÂNCIA DO MODELO	18
FIG 3. 7 : 5º INSTÂNCIA DO MODELO	18
FIG 3. 8: 6º INSTÂNCIA DO MODELO.....	19
FIG 3. 9: MODELO DE FOLHA DE CÁLCULO AVANÇADO	20
FIG 3. 10: INSTÂNCIA DO MODELO DE FOLHA DE CÁLCULO AVANÇADA.....	20

Lista de siglas

MDE - Model-Driven Engineering

UML . Unified Modeling language

SSRB - Spreadsheet Standards Review Board

Capítulo I

1. Introdução

Durante anos, a estrutura tabular foi muito usada para organizar dados. A figura 1.1 mostra um exemplo da estrutura de dados tabular, a tablete de Plimpton 322, datada por volta de 1800 BC [10]. A tablete Plimpton 322 é um exemplo de uma tabela que contém 4 colunas e 15 linhas com dados numéricos. Para cada coluna existe um cabeçalho descrito e 4 colunas que contêm a numeração das linhas de 1 a 15, escrito em sistema numeração babilônico [4].



Fig 1. 1:Plimpton 322 . uma tablete datada de volta de 1800 BC

A estrutura tabular traz grandes vantagens, permite análise sistêmica da informação e ajuda a estruturar valores para realizar cálculos.

1.1 Folhas de cálculo

O termo folhas de cálculo e folha de trabalho tiveram origem antes da existência da folha de cálculo eletrônica. Ambas, têm o mesmo significado. Mas, o termo folha de trabalho foi usado, primeiro, em 1970. Depois em 1978 foi usado o termo folha de cálculo. Este termo, folhas de cálculo, veio a significar tabela de dados organizada em colunas e linhas.

Inicialmente, antes de emergir os computadores comerciais, as folhas de cálculo eram criadas em papel, com lápis e papel. As folhas em papel eram usadas somente para criar aplicações financeiras e comerciais.

Os benefícios das folhas de cálculo em papel fez com elas fossem usadas em vários domínios, nomeadamente, análise estatísticas, ou qualquer finalidade que requer entrada de dados ou realização de cálculos.

	Nº	Nome	Matemática	L.Portuguesa	Obs
1	AE1	Ana Paula	10	13	Apto
2	AE2	José Maria	8	9	N.Apto
3	AE3	Paulo João	12	10	Apto
4	AE4	Carlos Manuel	9	11	N.Apto

Fig 1. 2:Folha de cálculo em papel

A figura 1.2 ilustra um exemplo simples de uma folha de cálculo em papel utilizada para guardar as notas finais dos alunos. Esta folha de cálculo possui 5 colunas e 5 linhas, e os professores devem preencher as células vazias conforme os campos, dados dos alunos a as notas das disciplinas.

Com aparecimento dos computadores pessoais, surgiu vários tipos de folhas de cálculo electrónicas. Em 1979, surgiu a primeira folha de cálculo electrónica, chamada Visical, lançada pela Apple II para computadores pessoais. Este sistema de folha de cálculo Visical foi um sucesso, incluí uma interface textual composta por células. Mas, em 1984, foi lançado o Lotus, um sistema que destronou o Visicalc, que tinha melhor desempenho e inclui interface amigável e gráficos genéricos. Em 1990, foi lançado com sistema operativo Windows, o Excel. Actualmente, o Microsoft Excel tem a maior cota do mercado, é o número 1 de vendas.

Desde então, tem acontecido novas actualizações do editor de folha de cálculo por cima da internet, nomeadamente, Microsoft Office 365, Google, LibreOffice Calc, entre outros.

Actualmente, a folha de cálculo electrónica tem sido preservada com o mesmo sistema que usava o Visicalc, os dados são apresentados de forma tabular e podem ser individualmente referenciado usando coordenadas. Cada coordenada é composta por uma coluna, que são identificadas por letras e uma linha, identificadas por números. Cada célula pode conter valores e fórmulas que podem referenciar uma célula pelas respectivas coordenadas, ou regras de células, instâncias [9].

1.2 Motivação

A folha de cálculo electrónica é uma ferramenta amplamente usada por utilizadores finais, que não são programadores profissionais. Elas dão suporte a milhões de programadores e não profissionais para os mais variados fins. Muitas organizações utilizam as folhas de cálculo para criar, guardar informações sobre a gestão da empresa.

Entretanto, o sucesso das folhas de cálculo deve-se ao facto delas serem flexíveis, intuitivas, com um esquema tabular bidimensional combinando operações de inserção e remoção de linhas e colunas.

Mas, as folhas de cálculo são muito propensas a erros. Alguns estudos mostram que 90% das folhas de cálculo criadas pelas organizações tem erros [18]. Erros com taxas alarmantes que causam impacto negativo na gestão de empresas. Os principais erros são de digitação da informação, fórmulas e mudança da estrutura das tabelas.

Para minimizar o problema de erros, a comunidade de investigadores e vendedores de ferramentas de folhas de cálculo trabalharam vários anos e propuseram técnicas e metodologias, tal como, melhores práticas, teste e modelagem da folha de cálculo.

Em 2002, baseado na experiência e práticas de engenharia de software anteriores, Grossmam[12] apresentou as melhores práticas, directrizes [12]. Adicionalmente, FAST[11]e SSRB[13] publicaram a padronização de processos para engenharia da folha de cálculo.

Recentemente, Grossmam e Ozluk[12] escreveram 3 metodologias de melhores práticas para engenharia da folha de cálculo. Eles acreditam que cada metodologia de folha de cálculo melhora a produtividade, a precisão e a capacidade de manutenção de grandes modelos de planeamento financeiros.

O uso das melhores práticas de engenharia da folha de cálculo ajuda o utilizador a criar folhas de cálculo com menos erros. Mas, não garante que as folhas de cálculo estejam livres de erros.

Com decorrer do tempo, foi adoptada técnicas de verificação da folha de cálculo, tal como, técnica de verificação semântica, técnica de teste de auditoria e teste de formatação. A técnica de verificação consiste em testar a semântica da folha de cálculo, os valores de entrada, as fórmulas e os seus resultados, conteúdo da célula correspondem com os conteúdos esperados. O teste de formatação garante que a folha de cálculo segue um predeterminado conjunto de boas práticas. A técnica de teste de auditoria consiste em verificar todas as fórmulas e valores calculados na folha de cálculo.

Contudo, olhando as iniciativas, os investidores de folhas de cálculo, verificaram através de estudos, que o desenvolvimento da folha de cálculo não tinha em conta os princípios de engenharia de software, muito usado por profissionais de desenvolvimento de software[1].

Para superar o problema de erros constantes, a comunidade de investigação de folha de cálculo sentiu a necessidade de aplicar metodologias de

Engenharia orientada por modelos (MDE). Usando esta metodologia é possível o utilizador detectar erros e prevenir e impor restrições no desenvolvimento do software. A engenharia orientada por modelos (MDE) é uma metodologia de desenvolvimento de software que usa modelos de domínio ou representação abstracta de software. Eles propuseram, baseado em MDE, orientação por objectos, técnicas, tal como, ClassSheets, diagrama de UML e templates de folhas de cálculo para diminuir os erros. Usando estas técnicas o utilizador pode especificar modelos de folhas de cálculo mais facilmente.

1.3 Objectivo

O objectivo deste trabalho é especificar as funcionalidades avançadas da folha de cálculo usando o ClassSheets descrito em [1], [3] e [6].

Com este trabalho, pretende-se responder às seguintes perguntas:

São as extensões existente para ClassSheets suficientemente expressivas para modelar as folhas de cálculo reais?

Que novas extensões são necessárias para usar ClassSheets num ambiente real?

As extensões são mais facilmente utilizadas por programadores e não programadores?

1.4 Estrutura da dissertação

Este trabalho está organizado da seguinte maneira:

Capítulo II será apresentado a abordagem orientada por modelos, nomeadamente, templates de folha de cálculo e ClassSheets. Depois, vamos ilustrar a especificação do modelo de folha de cálculo usando os ClassSheets.

Capítulo III será apresentado a proposta de botão recursivo, o modelo e as várias etapas para descrever a instância de modelo da folha de cálculo. No final, vamos apresentar a especificação do modelo de folha de cálculo avançado.

No capítulo IV, será apresentado as conclusões do trabalho.

Capítulo II

2. Folhas de cálculo dirigidas por modelos

Na tentativa de resolver o problema de erros da folha de cálculo os investigadores decidiram aplicar várias técnicas baseadas na abordagem orientada por modelos (MDE), tal como, Templates de folha de cálculo [14], os ClassSheets[1] e diagrama de classes[8]. O MDE é um conjunto de metodologias de desenvolvimento de software que usa abstracção através da modelagem para especificar peças de software. Usando as técnicas propostas é possível o utilizador executar com segurança etapas de edição das suas folhas de cálculo e introduzir uma nova forma de desenvolvimento da folha de cálculo orientado por modelos[1].

Na secção 2.1 e 2.2 vamos apresentar os Templates de folha de cálculo [14] e os ClassSheets descritos em [1], [3] e [6]. Depois, vamos ilustrar a especificação do modelo de folha de cálculo usando os ClassSheets. O objetivo é estudar o nível de expressividade de modelo de ClassSheets num contexto real. Os ClassSheets são a base deste trabalho.

2.1 Templates de folha de cálculo

Os templates de folha cálculo é um formalismo, proposto por Abraham e Erwig[14], que permite definir o conteúdo e estrutura da folha de cálculo. Usando os templates de folha de cálculo é possível criar aplicações de folhas de cálculo similares as aplicações concretas com menos erros.

As especificações do template de folhas de cálculo são construídas com editor VITSL e armazenadas dentro Gencel[14]. A ferramenta Gencel é uma extensão de Excel que fornece um ambiente onde as operações são actualizadas dependendo da especificação fornecida. Usando a ferramenta Gencel é possível testar os erros de templates da folha de cálculo e gerar a instância da folha de cálculo concreto automaticamente com menos erros.

Os templates de folha de cálculo podem ser representados textualmente e visualmente. Para estruturar folhas de cálculo, Abraham e Erwig propuseram o VITSL, uma linguagem visual que permite reusar e prevenir erros [14]. A linguagem VITSL fornece métodos para modelagem dos templates de folha de cálculo e maneira de envolvê-la.

O VITSL oferece construtores para definir tabelas com sequência horizontal fixa e colunas estendidas, onde a coluna é construída com sequência vertical

fixa e blocos estendidos, que são conjuntos rectangulares de células contém valores e fórmulas.

Na figura 2.1, consideramos um exemplo simples do Template de folha de cálculo retirado[14]. Este Template de folha de cálculo descreve a estrutura de uma coluna de números, inclui o cabeçalho em cima, chamado Values e uma fórmula de somatório no rodapé. O Template da tabela pode ser especificado usando a seguinte expressão.

	A
1	Values
2	0
.	
.	
.	
3	= Sum(A2)

Fig 2. 1: Template da folha de cálculo Pilotos

A especificação consiste em 3 elementos: 2 células, a etiqueta do cabeçalho, a célula do rodapé que contém a fórmula da soma e o grupo de expansão vertical, que consiste numa célula simples que contém o valor 0. O argumento da fórmula do somatório, A2, é mostrado como um endereço absoluto nesta especificação.

O VITSL oferece os seguintes elementos visuais para descrever os Templates:

Células, representadas por rectângulos e compostas com dados e fórmulas.

Referências, representadas por endereços da célula.

Grupos Verticais, representados por pontos verticais que indicam a possível expansão de um grupo de célula (s).

Grupos Horizontais, representados por pontos horizontais que indicam a possível expansão de uma ou mais colunas na direção horizontal.

VITSL é uma linguagem visual muito similar a %standard+da folha de cálculo. A similaridade garante alta aceitação do VITSL pelos utilizadores tradicionais. A sintaxe VITSL foi formalmente definida[14]. Os elementos da sintaxe estão resumidos na figura 2.2.

$f \ ? \ Fml ::= v \ \ \psi(f, \dots, f) \text{ (fórmulas)}$ $b \ ? \ Block ::= f \ \ b \ \ b \ \ b^*b \text{ (blocos, tabelas)}$ $c \ ? \ Col ::= b \ \ b^* \ \ c^*c \text{ (colunas)}$ $t \ ? \ Template ::= c \ \ c^* \ \ t \ \ t \text{ (templates)}$
--

Fig 2. 2: Sintaxe do Templates VITSL

Os Templates VITSL [14] são constituídos por blocos (b), que podem conter fórmulas (f). Os blocos dos Templates têm uma representação básica, a célula, ou podem ter outros blocos, formando colunas (c). Entretanto, o bloco do Templates (b ou c) pode ser expandidos, ou seja a sua instância pode ser repetida na vertical ou na horizontal. Quando um bloco representa uma célula, este contém valores básicos, por exemplo texto e números, uma referência, ou uma expressão construída com fórmulas, que varia com o número de argumentos.

Além da sintaxe abstracta livre de contexto da VITSL dada pelo BNF, também foi definido um sistema do tipo para permitir a restrição e a construção de fórmulas e Templates. Os sistemas do tipo impõem o correcto alinhamento de linha e colunas em relação a repetição do grupo. É garante a correcção de referências dentro de tabelas e fórmulas de agregação e a correcta digitalização de expressões e fórmulas. Em particular, as restrições de alinhamento garantem inserção e exclusão operações de linhas ou colunas.

Entretanto, os templates de folhas de cálculo fornecem uma maneira de garantir a especificação do esquema e preencher as células mais facilmente. Mas, o template de folha de cálculo ainda não tem um interface amigável, as células usam referências e o conteúdo do template não tem um modelo de negócio associado entre si. Para resolver alguns problemas, foi desenvolvido outro modelo, chamado modelo de ClassSheets

2.2 ClassSheets

Os ClassSheets são um formalismo de alto nível, orientado por objecto para especificar a lógica do negócio da folha de cálculo [1]. O ClassSheets permite o utilizador expressar a estrutura e a lógica de negócio de folhas de cálculo usando o conceito de UML, classes e atributos. Usando o modelo de ClassSheets é possível definir tabelas de folha de cálculo, definir etiquetas para colunas das tabelas, especificar o tipo de valores que as colunas podem conter, dar nomes as tabelas e a maneira de expandir tabelas, horizontalmente ou verticalmente.

O ClassSheets define folhas de cálculo contendo as classes (c), formado por blocos (b). Ambos, as classes e blocos podem ser expandidos, as suas instâncias podem ser repetidas tanto na horizontal (c_{\rightarrow}) e na vertical (b_{\downarrow}). As classes são identificadas por etiquetas. Um bloco pode ser representado, na

sua forma base, em células da folha de cálculo ou pode ser uma composição de outros blocos. Ao representar uma célula, um bloco pode conter um valor base(, por exemplo um número inteiro), ou um atributo(A=F), que é composto por um nome de atributo(um) e um valor (f). O atributo pode definir três tipos de células : 1) um valor de entrada, que dá essa indicação, 2) uma referência chamada por outro atributo (NA), em que n é o nome da classe e o nome de atributo) ou 3) uma expressão construída através da aplicação de funções para um número de variáveis de argumento fornecido por uma fórmula((f,...,M)).

A ClassSheets tem uma sintaxe abstracta que está resumida na figura 2.3. A sintaxe permite representar ClassSheets textualmente [1]. Além da representação textual, o ClassSheets tem uma representação visual, como descrito mais a abaixo.

$$\begin{array}{l}
 f \in Fml ::= \varphi \mid n.a \mid \varphi(f, \dots, f) \quad (\text{formulas}) \\
 b \in Block ::= \varphi \mid a = f \mid b|b \mid b^b \quad (\text{blocks}) \\
 l \in Lab ::= h \mid v \mid .n \quad (\text{class labels}) \\
 h \in Hor ::= \underline{n} \mid \underline{n} \quad (\text{horizontal}) \\
 v \in Ver ::= |n \mid \underline{n} \quad (\text{vertical}) \\
 c \in Class ::= l : b \mid l : b^b \mid c^c \quad (\text{classes}) \\
 s \in Sheet ::= c \mid c^{\rightarrow} \mid s|s \quad (\text{sheets})
 \end{array}$$

Fig 2.3:Sintaxe da ClassSheets. Fonte[1]

A sintaxe da ClassSheets define uma classe como uma composição de blocos. Um bloco na sintaxe é uma célula de uma ClassSheets. Os blocos podem ser organizados horizontalmente ou verticalmente e compõem o esquema da classe[2].

Para ilustrar o modelo de ClassSheets na prática, consideramos o sistema de marcação de voos duma companhia aérea retirada de [3]. Assumindo que qualquer linha aérea deve registrar actividade dos pilotos, típico de um sistema de software. A maneira simples de resolver o problema é usar a folha de cálculo, como mostra tabela da figura 2.4 a). Esta tabela tem um título, Pilots, e linha com etiquetas, uma para cada coluna da tabela: ID representa o identificador do piloto, o Name representa o nome do pilots e Age representa a idade de piloto. Cada linha subsequente representa o piloto de concreto. As tabelas, tal como, apresentadas na figura 2.5 a) são frequentemente usadas dentro da folha de cálculo.

Para ilustrar o modelo ClassSheets associado a tabela Pilots, vamos considerar, a figura 2.4 b) que ilustra um exemplo do modelo de ClassSheets da tabela Pilotos.

A classe Pilots tem uma etiqueta simples (fixa) na coluna A1. E cada piloto tem três atributos: ID, Name, Age. e que corresponde a 3 etiquetas que ficam

na linha, A1, A2, A3. O valor que a célula pode conter é especificado na célula por baixo: duas strings com valores por defeito %t e um número (com valor inicial 0).

Na primeira coluna, a linha 4 contém o símbolo com 3 pontinhos verticais. Isto indica que a classe Pilots pode crescer verticalmente.

Na figura 2.4b), a instância que conforma com modelo. As linhas 1 e 2 contêm etiquetas definidas no modelo e as linhas 3 a 5 contêm informação de 3 pilotos em que as células incluem valores que correspondem valores especificado pelo modelo. No modelo as células coloridas mostram que todos são parte da mesma tabela.

	A	B	C
1	ID	Name	Age
2	p1	Jonh	45
3	p2	Mike	39
4	P3	Anne	56

(a) Tabela Pilots

	A	B	C
1	Pilots		
2	ID	Name	Age
3	id=""	name=""	age=0
⋮			

(b) Modelo de ClassSheets visual Pilots

Fig 2. 4: Exemplo Pilotos

O ClassSheets tem uma sintaxe abstrata que permite definir o ClassSheets textualmente. A sintaxe da ClassSheets não captura estruturas que resultam da organização 2-dimensional. Para certificar se as ClassSheets são bem formadas foi apresentada o outro conjunto de regras chamadas, regras de tilling.

Um tilling é definido de acordo a sua agregação de tilling. O tilling define 4 tipos de estruturas de tilling, nomeadamente, classes simples sem agregação, classe agregada unidimensional, classes agregadas horizontalmente e verticalmente e agregações bidimensional. As agregações de tilling pode ser principalmente aninhada, e o tilling pode ser construído com uma composição vertical ou horizontal [1]. O tilling dá-nos a informação sobre a construção da classe e a sua composição.

O ClassSheets contém uma semântica que permite descrever os tipos de dados nos modelos de ClassSheets. Mas, a semântica de ClassSheets é pobre [6], apenas descreve tipo de dados simples, não permite definir tipos de dados complexos e relacionar atributos. Foi neste contexto que Cunha et. al. [6] propuseram a extensão semântica de modelo ClassSheets. Para melhorar a semântica do modelo de ClassSheets foi apresentado 3 tipos de técnicas, nomeadamente, modelos relacionais, expressões regulares e tipos de restrições. A ideia é guiar o utilizador a introduzir dados correctamente e evitar corrupção dos dados da folha de cálculo.

O ClassSheets define uma representação visual baseado no template de VITSL proposto por Abraham [14], muito próximo da folha de cálculo

concreta[1]. A representação visual do ClassSheets foi embebida num sistema de folhas de cálculo. O embebimento do modelo visual num ambiente de folha de cálculo foi proposto por Cunha et. al.[3] e garante que utilizador edita com segurança folhas de cálculo. Tal embebimento, por definição, fornece ao utilizador final um ambiente de desenvolvimento MDE de folhas de cálculo num contexto de desenvolvimento que sincroniza o modelo de ClassSheets e a respectiva instância. Para embeber o modelo de Classsheets num sistema de folhas de cálculo a linguagem de ClassSheets foi modificada [3].

2.3 Especificação do modelo da folha de cálculo usando o ClassSheets

Nesta seção, vamos apresentar a especificação do modelo de folhas de cálculo usando os ClassSheets proposto por Engels e Erwig[1] e Mendes na sua tese de Doutoramento. Pretende-se investigar as extensões existentes do modelo de ClassSheets são suficientes expressivas para modelar folhas de cálculo usadas no contexto real.

Exemplos:

Para melhor ilustrar o trabalho realizado na prática, nesta dissertação, consideramos a figura 2.5 - mostra a instância da tabela da Análise de Financeira de custo, adaptada, usada por investigadores de folhas de cálculo para teste e o respectivo modelo de ClassSheets mostrado na figura 2.6.

	A	B	c	D
1	Análise Financeira de Custos	Ano - 2001	Ano - 2002	Ano - 2003
2	Custos			
3	Caixas de cigarros			
4	Restaurante	10	20	10
5	Cigarro Tobacco	20	15	5
6	Caixa de cigarros Total	30	45	15
7	Custo Total	30	45	15

Fig 2. 5: Instância da tabela de Análise financeira de custos, da folha de cálculo adaptada, ficheiro `financial/processed/costfactors.xlsq` do corpus de folhas de cálculo EUSES [17].

Na figura 2.5, consideramos um exemplo de folha de cálculo, retirada da base de dados criada pelos investigadores para teste, nomeadamente o ficheiro `financial/processed/costfactors.xlsq` do corpus de folhas de cálculo EUSES [17]. Esta tabela contém informação da Análise Financeira de custos empresa referente a três anos.

A tabela, figura 2.5,têm quatro colunas, nomeadamente, A, B, C e D, e oito linhas, nomeadamente as linhas 1 a 7 A junção das linhas e as colunas

temos as células. As células são definidas por etiquetas. Na coluna A, temos a informação do título da tabela, a coluna B, C e D cada uma tem definido o ano que ocorreu a análise do custo. Por baixo da linha 1, temos definido os custos e os itens dos custos, que podem ter ou não subitens. Na linha 7, temos o total do 1º item. No final, a linha 7, temos a informação referente ao total dos custos. Associados os etiquetas dos itens, subitens temos uma lista de valores e os totais dos itens e custos realizados na empresa durante 3 anos.

	A	B	...	D
1	Análise Financeira de Custos	Ano-2001		
2	Custos			
3	Itens= " "	values = 0		
⋮				
5	ItensTotal= " "	itenstotal=Sum(values)		
6				
7	Custos Total	custosTotal=Sum(itenstotal)		

Fig 2. 6: Modelo de ClassSheet Análise Financeira de Custos

A tabela Análise financeira de custos, figura 2.5, pode ser modelada visualmente como mostra figura 2.6 usando a notação/sintaxe do modelo de ClassSheets descrito por Engel et. al. [1] e as extensões do modelo descrito por Jorge Cunha na sua tese de Doutorado.

O modelo de ClassSheets, figura 2.6, é constituído por 4 classes, nomeadamente, classe análise financeira de custos, classe Custo, classe itens e classe Ano.

A classe Análise financeira de custos que tem a etiqueta na coluna A1, tem toda a informação, incluindo: informação sobre custos, os itens de custo e o ano, o total dos itens por ano e o total dos custos por ano.

A classe Itens que tem a etiqueta na coluna A3, é composta por 3 atributos, nomeadamente, itens e values e total dos itens. Os valores totais dos itens são agregados na classe ano e custo. A classe Custo agrega a soma de todos Custos.

A classe Custo que tem a etiqueta na coluna A2, pode ser expandida na vertical através dos atributos itens e values, enquanto a classe ano pode ser expandida horizontalmente usado o atributo ano e values.

A classe Analise financeira de custo que agrega a classe Custo e classe Ano. A classe Ano que tem a etiqueta na coluna B1, agrega o atributo ano. A classe itens é uma classe de relacionamento, e está agregada na classe Ano e Custo.

Entretanto, observando os padrões do modelo de ClassSheets, figura -2.6, e a folha de cálculo, figura 2.5, que mostra a tabela da análise financeira de custo. Conclui-se que os ClassSheets tradicionais permitem representar tabelas simples sem etiquetas composta, ligadas entre si hierarquicamente. Portanto, analisando folha de cálculo, acima, figura 2.5 observa-se uma estrutura de tabela com etiquetas compostas por etiqueta. A principal etiqueta representa a categoria, e acopla etiquetas pertencentes a categoria principal onde estão ligadas, entre si.

2.4 Resumo

Neste capítulo, revisamos os templates de folha de cálculo, um modelo de folha de cálculo que permite gerar aplicações de folhas de cálculo. De seguida, descrevemos ClassSheets, um formalismo baseada na abordagem MDE e OO. Os ClassSheets são a base deste trabalho.

Adicionalmente, apresentamos a especificação do modelo de folhas de cálculos usando os ClassSheets e a respectiva instância. Com a especificação do modelo de folha de cálculo pretende-se medir a expressividade do modelo de folha de cálculo usando ClassSheets para representar folhas de cálculo num contexto real.

Capítulo III

3. Modelo de folha de cálculo Avançado

O ClassSheets[1] é formalismo de alto nível muito poderoso, que permite descrever e estruturar a lógica do negócio da folha de cálculo. Usando os modelos ClassSheets é possível melhorar, definir de forma abstracta a estrutura de folha de cálculo, sua instância, para possibilitar a derivação inserção de dados reais.

Neste capítulo, primeiro, vamos ilustrar o botão recursivo, um mecanismo que permite especificar folhas de cálculo usadas num contexto real. De seguida, vamos apresentar o modelo de folha de cálculo e as várias etapas para gerar a instância do modelo. Este modelo de folha de cálculo será descrito com base a linguagem ClassSheets tradicional e o botão recursivo. No final, vamos discutir resultados obtidos.

3.1 Mecanismo recursivo

Nesta secção, vamos definir um novo mecanismo, chamado botão recursivo, proposto para estender o modelo de ClassSheets e resolver o problema de insuficiência do ClassSheets.

	A
1	String = " "
2	.../R

Fig 3. 1: Botão recursivo.

A figura 3.1- mostra o botão recursivo, um mecanismo que permite expandir itens recursivamente ou verticalmente. Usando este mecanismo, o utilizador pode associar itens compostos e ligar hierarquicamente entre si.

Este mecanismo tem uma estrutura rectangular. É constituída por 2 blocos de células, a célula com botão recursivo e célula com atributo simples. O atributo define os dados da tabela, uma string vazia, o botão recursivo define duas operações, ã /R, as 3 reticências permite o utilizador expandir os dados da string para vertical, a letra R permite o utilizador também expandir os dados do campo do atributo string de forma recursiva.

Para melhor ilustrar o novo mecanismo, o botão recursivo, na prática, vamos de seguida apresentar a várias etapas para gerar o modelo de folha de cálculo avançado e a sua instância.

3.2 Validação do botão recursivo

O botão recursivo é um mecanismo que possibilita o utilizador descrever folhas de cálculo com itens compostos, ligados entre si hierarquicamente. Entretanto, este mecanismo foi definido com base as especificações realizadas nas folhas de cálculo usadas num contexto real, retirada da base de dados dos investigadores de folha de cálculo usada para teste.

Nesta seção pretende-se avaliar o novo mecanismo, o botão recursivo. Para tal, vamos primeiro especificar o modelo de folha de cálculo usando o botão recursivo e a linguagem de ClassSheets, a notação visual. Depois, com base a especificação do modelo de folha de cálculo, figura 3.2, vamos gerar as várias instâncias de modelo de ClassSheets.

Para criar o modelo de folha de cálculo usando ClassSheets e botão recursivo, considerando a folha de cálculo ilustrada na figura 2.5, pode-se:

- Adicionar uma classe para análise financeira de custos, seleccionamos a célula A1;
- Adicionar uma classe para o ano, seleccionado a célula B1 e escolhemos a classe para expandir horizontalmente;
- Adicionar uma classe para o custo, seleccionado a célula A2 e escolhemos a classe para expandir horizontalmente;
- Adicionamos a classe itens e atribuímos o seu valor de defeito, e agregamos o atributo valor e colocamos o seu valor por defeito. E escolhemos para expandir verticalmente;
- Adicionamos o total dos itens, seleccionamos a célula B5 e total dos custos, seleccionamos a célula B6 e respectivas etiquetas;
- Adicionamos o botão .../R , seleccionamos o intervalo A1:A2;
- Depois de cada passo para criação do modelo, manualmente, é possível mostrar o resultado da co-evolução dos dados, de modo que utilizador tenha uma ideia precisa dos efeitos resultantes da ação;

Para gerar as várias instâncias do modelo de folha de cálculo, a partir do modelo apresentado na figura 3.2. Temos as seguintes etapas:

1º: Etapa - Definir modelo de folha de cálculo com itens compostos usando o botão recursivo de acordo a folha de cálculo concreta.

	A	B	C	D
1	Análise Financeira de Custos	Ano-2001		
2	Custos			
3	Itens= " "	values = 0	.	
4		.../R	.	
5	ItensTotal= " "	itenstotal=Sum(values)	.	
6				
7	Custos Total	custosTotal=Sum(itenstotal)		

Fig 3. 2: Modelo de folha de cálculo avançado.

A figura 2.5- mostra a folha de cálculo Análise Financeira de Custos, e a figura 3.2 . apresenta a especificação do modelo da folha de cálculo da Análise Financeira de Custos.

Este modelo foi construído com 4 classes, nomeadamente, classe análise financeira de custos, classe Custo, classe itens e classe Ano, representada numa estrutura tabular bidimensional. A classe Análise financeira de custo agrega 2 classes, nomeadamente, Custos e Ano. A classe Itens está agregada a classe Custo e Ano.

A classe Análise financeira que tem a etiqueta na coluna A1 e tem associado 1 atributo: CustosTotal e que corresponde a 1 etiqueta que fica na célula A7. O valor que célula pode conter é especificado na célula do lado, B7. O custosTotal define a soma de todos custos.

A classe Custo que tem a etiqueta na coluna A2, tem associado 1 atributo: itenstotal e que corresponde a 1 etiqueta que fica na célula A5. O valor que célula pode conter é especificado na célula do lado, B5. O itensTotal define a soma de todos itens de custos.

A classe Ano que tem a etiqueta na coluna B1, tem associado 2 atributos: ano, itenstotal e que correspondem a duas etiquetas que ficam na célula B1, A5. O valor que célula pode conter é especificado na célula B5, B1. O itensTotal define a soma de todos itens de custos.

A classe Itens que tem a etiqueta na coluna A3, tem associado 2 atributos: itens e valor e que corresponde a 2 etiquetas que ficam na célula A3, B3. O valor que células podem conter é especificado na mesma célula, A3, B3. Os itens definem o tipo de itens da lista, o valor define a quantidade dos itens.

No lado direito, a cor vermelha da coluna D indica que a classe itens pode se expandir verticalmente ou recursivamente, usando as funcionalidades do botão recursivo, ilustrado na linha 4. Os 3 pontinhos definidos na coluna C indicam que a classe ano pode crescer horizontalmente.

2º: Etapa - Gerar a instância do modelo

	A	B	C	D
1	Análise Financeira de Custos	Ano-2001		
2	Custos			
3	" "	0	.	
4	.../R		.	
5	" "	0	.	
6				
7	Custo Total	0		

Fig 3. 3: 1.^a Instância do Modelo

A figura 3.3 - mostra a 1.^a instância do modelo de folhas de cálculo. A 1.^a instância foi gerada através do modelo de folha de cálculo apresentado na figura 3.2. Para gerar a 1.^a instância do modelo definiu-se duas string vazias, célula A3 e A5, e os valores por defeitos, célula B3 e B5. Associado a célula B3 e temos o Botão recursivo, linha 4. A string vazia, célula A3, é descrita em função a etiqueta de custos.

Usando botão recursivo(.../R) é possível expandir o bloco verticalmente/recursivamente. E calcular automaticamente o valor dos subitens, célula B5 e o total de todos os custos, célula B7.

Entretanto, do lado direito da tabela, figura 3.3, na coluna D, temos um rectângulo de cor vermelha que indica a expansão do bloco, verticalmente/recursivamente. Usando a eclipse da coluna C é possível definir a expansão do bloco horizontalmente e incluir vários anos.

3º: Etapa . editar a string vazia

	A	B	C	D
1	Análise Financeira de Custos	Ano - 2001		
2	Custos			
3	"Caixa de cigarros "	0	.	
4	.../R		.	
5	"Caixa de cigarros Total"	0	.	
6				
7	Custo Total	0		

Fig 3. 4: 2.^a Instância do modelo

A figura 3.4, mostra a edição do item caixa de cigarros associado a etiqueta custos, e a etiqueta Caixa de cigarros Total da tabela Análise financeira de custos.

Para editar o novo item, vamos adicionar na célula A3 o nome Caixa de cigarros, e inserimos na célula A5 o nome o Caixa de cigarros Total. Associado ao bloco A3 e B3 temos o botão recursivo. Usando este botão

recursivo é possível expandir o item Caixa de cigarros e calcular automaticamente o valor dos subitens, célula B5, e o total de todos os custos, célula B7.

Entretanto, no lado direito da tabela, o rectângulo vermelho indica a expansão do botão ã /R1 (verticalmente/ recursivamente) associado ao item caixa de cigarros. Usando a eclipse da coluna C é possível definir a expansão do bloco horizontalmente e incluir vários anos.

4º: Etapa - Clicar R.

	A	B	C	D	E
1	Análise Financeira de Custos	Ano - 2001			
2	Custos				
3	"Caixa de cigarros "	0	-		
4	" "	0			
5	... / R2		-		
6	... / R1				
7					
8	"Caixa de cigarros Total"	0			
9					
10	Custo Total	0			

Fig 3. 5: 3.^a Instância do modelo

A figura 3.5 . mostra a 3.^a instância do modelo de folha de cálculo. Esta instância mostra expansão do item Caixa de Cigarros da tabela da Análise Financeira de Custos usando o botão recursivo.

Para expandir o item e criar um subitem foi criado uma string vazia que define a expansão do item recursivamente e 2 botões, o botão ã /R1 e o Botão ã /R2. Usando o Botão ã /R1 e ã /R2 é possível expandir o item caixa de Cigarros e subitem verticalmente/recursivamente.

Entretanto, no lado direito da tabela, o rectângulo vermelho indica a expansão do botão ã /R1 (verticalmente/ recursivamente) associado a item caixa de cigarros, célula A3. O rectângulo amarelo indica a expansão do botão ã /R2 (verticalmente/ recursivamente) associado ao novo item, célula A4. Usando a eclipse da coluna C é possível definir a expansão do bloco horizontalmente e incluir vários anos.

5º: Etapa - Editar string

	A	B	C	D	E
1	Análise Financeira de Custos	Ano - 2001			
2	Custos				
3	"Caixa de cigarros "	0	.		
4	"Restaurante "	10	.		
5	... / R2		.		
6	... / R1		.		
7			.		
8	"Caixa de cigarros Total"	10			
9					
10	Custo Total	10			

Fig 3. 6: 4ª Instância do modelo

A figura 3.6 . apresenta a 4ª instância do modelo de folha de cálculo. Esta instância mostra a edição do subitem do item Caixa de cigarros com o nome Restaurante e o valor a 10. Com a edição do valor do item restaurante a 10, o valor Custo total e o Caixa de cigarros total são alterados automaticamente com mesmo valor.

Entretanto, no lado direito da tabela, temos 2 rectângulos que indicam informação de expansão dos blocos (itens) (verticalmente/recursivamente). Associado ao item restaurante temos o botão \odot /R2 e rectângulo vermelho e ligado ao item caixa de cigarros temos o botão \odot /R1 e rectângulo amarelo. Usando a eclipse da coluna C é possível definir a expansão do bloco horizontalmente e incluir vários anos.

6º: Etapa - Clicar ...do botão .../R2

	A	B	C	D	E
1	Análise Financeira de Custos	Ano - 2001			
2	Custos				
3	"Caixa de cigarros "	0	.		
4	"Restaurante "	10	.		
5	" "	0	.		
6	... / R2		.		
7	... / R1		.		
8			.		
9	"Caixa de cigarros Total"	10			
10					
11	Custo Total	10			

Fig 3. 7: 5.ª Instância do modelo

A figura 3.7 . apresenta a 5.ª instância do modelo de folha de cálculo. Esta instância mostra na célula A5 e B5 a expansão recursiva do item Caixa de cigarros usando o botão recursivo. Para expandir o item Caixa de cigarros foi criada uma string vazia, célula A5 e o valor por defeito, célula B5.

Entretanto, no lado direito da tabela, temos 2 rectângulos que indicam a expansão dos blocos (itens) (verticalmente/recursivamente). Associado ao item restaurante temos o botão \odot /R2 e o rectângulo vermelho, ligado ao item caixa de cigarros temos o botão \odot /R1 e rectângulo amarelo. Usando a eclipse da coluna C é possível definir a expansão do bloco horizontalmente e incluir vários anos.

7º: Etapa: Editar o novo item criado

	A	B	C	D	E
1	Análise Financeira de Custos	Ano - 2001			
2	Custos				
3	"Caixa de cigarros "	0	.		
4	"Restaurante "	10	.		
5	"Cigarro Tobacco "	20	.		
6	.../R2		.		
7	.../R1		.		
8			.		
9	"Caixa de cigarros Total"	30			
10					
11	Custo Total	30			

Fig 3. 8: 6.^a Instância do modelo

A figura 3.8 . mostra a 6.^a instância do modelo de folha de cálculo. Esta instância do modelo mostra a edição do novo item, cigarro tobacco, e o valor do item no ano 2001, célula A5, B5.Com a edição do valor do item restaurante, o valor da etiqueta Total caixa de cigarro e o Custo total são alterados automaticamente.

Entretanto, do lado direito da tabela, temos a informação que indica a expansão dos itens. Associado ao item restaurante e item Caixa de cigarros temos o botão \odot /R1 e ligado ao item Cigarro Tobacco temos o botão \odot /R2. Estes botões permitem expandir os itens verticalmente ou horizontalmente. Para ilustrar a expansão do item, definimos 2 cores, vermelha e amarela. A cor vermelha associa o item caixa de cigarros e o item Restaurante ao botão \odot /R1, a cor amarela está associada ao item cigarro tobacco e o botão .../R2. Usando a eclipse da coluna C é possível definir a expansão do bloco horizontalmente e incluir vários anos.

3.3 Modelo de folha de cálculo avançado

Nesta secção, vamos apresentar o modelo de folha de cálculo avançado e a respectiva instância, com um exemplo prático. Pretende-se analisar a expressividade do modelo de ClassSheets com base o novo mecanismo, o botão recursivo, usando a folha de cálculo apresentada na figura 2.5.

Para analisar a expressividade do modelo de ClassSheets num contexto real, consideramos a instância da tabela Análise financeira de Custos,

apresentada na figura 2.5 e o respectivo modelo de ClassSheets, figura 2.6. Por outro lado, consideramos o modelo de folha de cálculo avançado, apresentado na figura 3.10 e sua respectiva instância, figura 3.9.

Exemplo:

	A	B	C	D
1	Análise Financeira de Custos	Ano-2001		
2	Custos			
3	Itens= "	values = 0	.	
4		../R	.	
5	ItensTotal= "	itenstotal=Sum(values)	.	
6				
7	Custos Total	custosTotal=Sum(itenstotal)		

Fig 3. 9: Modelo de folha de cálculo avançado .

A figura 3.9, mostramos o modelo de folha de cálculo avançado, especificada a partir da folha de cálculo apresenta na figura 2.5. Este modelo de folha de cálculo descreve a informação da Análise Financeira de Custos de uma lista itens composto por subitem, ligados entre si. O modelo de folha de cálculo foi descrito usando os ClassSheets apresentado em[1] e o novo mecanismo, o botão recursivo

	A	B	C	D	E
1	Análise Financeira de Custos	Ano - 2001			
2	Custos				
3	"Caixa de cigarros "	0	.		
4	"Restaurante "	10			
5	"Cigarro Tobacco "	20			
6		... / R2	.		
7		... / R1			
8			.		
9	"Caixa de cigarros Total"	30			
10					
11	Custo Total	30			

Fig 3. 10: Instância do modelo de folha de cálculo avançada

A figura 3.10-mostra a instância do modelo de folha de cálculo avançada gerado a partir do modelo descrito na figura 3.9. Esta instância descreve uma tabela com uma estrutura tabular bidimensional que podem ser expandido horizontalmente e verticalmente/recursivamente.

Como podemos observar, a estrutura tabular da tabela da folha de cálculo, figura 2.5, é constituída por 3 blocos, nomeadamente, cabeçalho, corpo e rodapé. O cabeçalho tem uma estrutura simples, formada por etiquetas. O corpo tem uma estrutura de dados composta, constituído por etiquetas sobrepostas, ligadas entre si. O rodapé foi construído com fórmulas que permite calcular os resultados automaticamente.

Entretanto, considerando a figura 3.9 e 3.10 concluímos que, usando ClassSheets tradicionais é possível especificar folhas de cálculo com tabelas simples com expansão vertical, horizontal e ambas sem itens ligados entre e tabelas com representação tabular bidimensional. Para representar tabelas complexas com item compostos, os modelos tradicionais são insuficientes.

3.4 Resumo

Neste capítulo, foi proposto o botão recursivo, um mecanismo que permite a especificação do modelo de folhas de cálculos com itens composto, ligados entre si recursivamente.

Adicionalmente, ilustramos a validação de novo mecanismo, o botão recursivo. Para tal, descrevemos as várias etapas para especificar o modelo de folha de cálculo e a respectiva instância. A ideia é propor novas extensões de modelos tendo em conta as insuficiências do ClassSheets para modelar folhas de cálculo usada num ambiente real.

No final, ilustramos a especificação de modelo da folha de cálculo, com itens compostos usada num contexto real e respectiva instância, usando a linguagem ClassSheets e botão recursivo, e discutimos os resultados, comparando a especificação de modelo folha de cálculo com itens compostos e respectiva instância usando apenas a linguagem ClassSheets, apresentada na seção 2. A ideia é discutir o nível de expressividade das duas especificações de modelo de folha de cálculo num contexto real.

Capítulo IV

4. Conclusão

As folhas de cálculo são uma ferramenta usada por programadores e não programadores para representar a lógica do negócio. O ClassSheets é um formalismo, alto nível, orientado objeto que permite especificar a lógica e estrutura da folha de cálculo[1]. O ClassSheets define uma linguagem textual para descrever os modelos de folha de cálculo e uma linguagem visual, baseado na linguagem VITSL. Neste trabalho, foi analisado a insuficiência das extensões do modelo de ClassSheets. O resultado mostrou que os ClassSheets são insuficientes para modelar folhas de cálculo usadas contexto real. Para resolver o problema, propusemos e validamos um novo mecanismo, o botão recursivo. Para validar o botão recursivo, consideramos folhas de cálculo usadas num contexto real, especificamos o modelo de folha de cálculo e geramos as várias etapas da instância do modelo. Os resultados mostraram que usando os modelos de ClassSheets e o botão recursivo é possível modelar folhas de cálculo com tabelas com atributos aninhados. Os ClassSheets permitem modelar tabelas simples, verticais, horizontais e relacionais.

Capítulo V

5. Bibliografia

- [1] Gregor Engels, Martin Erwig: ClassSheets: automatic generation of spreadsheet applications from object-oriented specifications. ASE 2005: 124-133
- [2] Jan-Christopher Bals, Fabian Christ, Gregor Engels, Martin Erwig: ClassSheets - model-based, object-oriented design of spreadsheet applications. Journal of Object Technology 6(9): 383-398 (2007)
- [3] Jácome Cunha, João Paulo Fernandes, Jorge Mendes, João Saraiva: Embedding, Evolution, and Validation of Model-Driven Spreadsheets. IEEE Trans. Software Eng. 41(3): 241-263 (2015)
- [4] Jácome Cunha, João Paulo Fernandes, Jorge Mendes, João Saraiva: Spreadsheet Engineering. CEFP 2013: 246-299
- [5] Jácome Cunha, João Paulo Fernandes, Jorge Mendes, João Saraiva: MDSheet: A framework for model-driven spreadsheet engineering. ICSE 2012: 1395-1398
- [6] J. Cunha, J. P. Fernandes, J. Mendes, J. Saraiva, Extension and implementation of ClassSheet models, in: Proceedings of the 2012 IEEE Symposium on Visual Languages and Human-Centric Computing, VLHCC 12, IEEE Computer Society, Washington, DC, USA, 2012, pp. 19. 22.
- [7] Cunha, M. Erwig, and J. Saraiva, Automatically inferring ClassSheet models from spreadsheets, in 2010 IEEE Symposium on Visual Languages and Human-Centric Computing. IEEE Computer Society, 2010, pp. 93. 100.
- [8] Stevens, P., Whittle, J., Booch, G. (eds.): UML 2003 - The Unified Modeling Language, Modeling Languages and Applications, 6th International Conference, San Francisco, CA, USA, October 20-24, 2003, Proceedings, Lecture Notes in Computer Science, vol. 2863. Springer (2003)

- [9] Bricklin, D. VisiCalc: Information from its creators, Dan Bricklin and Bob Frankston. <http://www.bricklin.com/visicalc.htm> (last retrieved: 2012-09-01).
- [10] Robson, E. (2001). Neither Sherlock Holmes nor Babylon: A Reassessment of Plimpton 322. *Historia Mathematica*, 28(3):167206.
- [11] FAST (2010). The FAST Standard. <http://www.fast-standard.org/the-standard/> (last retrieved: 2012-09-01).
- [12] Grossman, T. A. and Ozluk, O. (2010). Spreadsheets Grow Up: Three Spreadsheet Engineering Methodologies for Large Financial Planning Models. *CoRR*, abs/1008.4174.
- [13] SSRB (2010). Best Practice Spreadsheet Modelling Standards (version 6.1). http://www.ssrb.org/best_practice_spreadsheet_modelling_standards.html (last retrieved: 2012-09-01).
- [14] Abraham, R., Erwig, M., Kollmansberger, S., and Seifert, E. (2005). Visual Specifications of Correct Spreadsheets. In *Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 05*, pages 189. 196. IEEE Computer Society.
- [15] Bruins, E.: On Plimpton 322. *Pythagorean Numbers in Babylonian Mathematics*. Koninklijke Nederlandse Akademie van Wetenschappen 52, 629. 632 (1949)
- [16] Hermans, F., Pinzger, M., van Deursen, A.: Automatically extracting class diagrams from spreadsheets. In: *ECOOP*. pp. 52. 75. Springer-Verlag (2010)
- [17] Marc Fisher and Gregg Rothermel. 2005. The EUSES spreadsheet corpus: a shared resource for supporting experimentation with spreadsheet dependability mechanisms. In *Proceedings of the first workshop on End-user software engineering (WEUSE I)*. ACM, New York, NY, USA, 1-5. DOI=<http://dx.doi.org/10.1145/1082983.1083242>
- [18] Panko, R. R. (2008). Spreadsheet errors: What we know. what we think we can do. *CoRR*,abs/0802.3457.