Universidade Nova de Lisboa
OMNIS CIVITAS CONTRA SE DIVISA NON STABIT
Faculdade de Ciências e Tecnologia

Departamento de Informática

MODELO PARA APOIO À DEFINIÇÃO CURRICULAR DE CURSOS 1º CICLO NA ÁREA DA CIÊNCIA DA COMPUTAÇÃO

Por

RUI PEDRO MARTINS LEAL

Tese submetida à Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, para cumprimento parcial dos requisitos para o grau de
Mestrado em Engenharia Informática

Orientador Científico: Prof. Dr.º João Moura-Pires

Monte de Caparica, Outubro 2010

Sumário

Pretende-se com esta dissertação, apresentar uma proposta de um modelo que suporte a definição e exploração de unidades curriculares e estruturas curriculares em cursos orientados à ciência da computação. Utilizando a informação disponível no *Computing Curricula* ACM e respectivo corpo de conhecimento (*CS Body of Knowledge*), bem como fontes internas no departamento de informática, este modelo permitirá obter várias formas de exploração desta informação, como por exemplo, métricas de comparação entre unidades curriculares, análises sobre a cobertura dos temas e das competências a obter nas unidades curriculares, etc.

É apresentada uma proposta de arquitectura recomendada para um eventual futuro sistema a implementar, bem como um protótipo exemplificativo com algumas das funcionalidades. Este permitirá a definição assistida de unidades curriculares individuais, estruturas curriculares, suportando o "corpo de conhecimento" ACM e a classificação de competências, bem como a definição e execução de relatórios com métricas simples para co-relacionamento da informação disponível. A arquitectura e o protótipo utilizarão diversos conceitos, como o mapeamento e criação de Ontologias, a representação em RDF e OWL, sistemas Web baseados em modelos de 3 camadas, RubyOnRails e ActiveRDF.

Abstract

The objective of this dissertation is to propose a data model for support the defintion and exploration of curricular units and structures of computer science programs. Using the information available in the *ACM Computing Curricula* and on the ACM-defined "CS Body of Knowledge", as well as several internal information sources available to the computer science department, this model is provide several ways to explore this information. As examples, curricular units comparison metrics, topic and skill coverage analysis on the curricular units, etc.

It is also presented a proposal for a recomended architecture to an eventual future concept implementation, as well as an example prototype with some of the funcionalities. This prototype will provide support for individual curricular unit definitions, curricular structures definition, making all the available “Body of Knowledge” as well as skills classifications. It will allow the definition and execution of reports with simple metrics for data co-relation. The architectura and the prototype will use different concepts like Ontology mapping and definition, data representation in RDF and OWL, 3-tier web-based systems, RubyOnRails and ActiveRDF.

Agradecimentos

À minha família, à Núria e aos meus amigos.

Ao meu orientador João Moura Pires, pela sua extensa colaboração nesta dissertação, pelos longos anos de colaboração profissional no qual sempre foi um exemplo e um guia, e pela sua amizade.

À Isabelina Jorge, igualmente pelas longos anos de colaboração profissional nos quais foi igualmente uma professora, guia, exemplo, referência de profissionalismo e integridade.

Aos meus colegas da UIDIU, Arlete Monteiro e Gilberto Gil, os quais partilharam comigo parte desta interessante experiência, não deixando de me motivar sempre.

À Dr.ª Laura Anna Ripamonti e sua equipa, pelo gentil fornecimento de informação útil para melhoria do Corpo de Conhecimento.

À equipa ITDS do projecto ECC, António Cruz, Pedro Rio e Filipe Caló, pela sua boa disposição e motivação.

Aos meus colegas de mestrado e colaboradores no projecto ECC, Ygor Cardoso, Ricardo Neves e Bernardo Oliveira, pelo apoio moral e excelente colaboração neste último ano.

Aos meus pais,
à Núria,
ao Jony e Isabelina
E os meus colegas UIDIU ...

Acrónimos

Acrónimo	Descrição
ECTS	European Credit Transfer System
CS	Computer Science
ACM	Association for Computing Machinery
IEEE	Institute of Electrical and Electronics Engineers
IEEE-CS	Institute of Electrical and Electronics Engineers Computer Society
ACM-CCS (ou CCS)	ACM Computing Classification System
ACM-CCS98 (ou CCS98)	ACM Computing Classification System – Revisão 1998
ACM-CS-BoK	ACM CS Body of Knowledge
CC2001	ACM/IEEE-CS Computing Curricula 2001
AIS	Association for Information Systems
CC2005	ACM/AIS/IEEE-CS Computing Curricula 2005
MADCC – CC ou MADCC	Modelo para Apoio à Definição Curricular de Cursos 1º ciclo na área da Ciência da Computação
BoK	Body of Knowledge
CdC	Corpo de Conhecimento
UC	Unidade Curricular
EC	Estrutura Curricular
PC	Plano Curricular
OC	Oferta Curricular
ACM-A	Conceitos no documento ACM CC2001 - Appendix A
ACM-B	Courses no documento ACM CC2001 - Appendix B
DI	Departamento de Informação - FCT-UNL
LOs	Learning Objectives (do ACM CC2001)
DDs	Dublin Descriptors (5)
DDs-e	Dublin Descriptors orientados à engenharia (7)

Tabela 1.1 - Lista de principais acrónimos

Índice

Índice de Figuras

Índice de Tabelas

Convenções

Notações utilizadas

Sublinhado refere para ligações dentro deste documento.

Negrito pretende dar ênfase a determinada frase ou expressão.

Itálico pretende indicar que se trata duma língua distinta do português.

"*itálico entre aspas*" indicam citações doutros trabalhos ou documentos.

Notas de rodapé pretendem esclarecer em detalhe algo adicional ao texto.

Uso de expressões em língua estrangeira

Sendo esta uma dissertação do curso de Eng.ª Informática, uma área onde a língua inglesa é utilizada como referência para os termos mais habitualmente utilizados, onde grande parte não tem uma tradução directa ou, quando existe, perde algum do sentido e expressividade original, optou-se pela utilização directa de alguns termos na língua original.

Principalmente, termos que refiram aspectos de caracterização da área e do ensino da mesma, ou termos e nomenclaturas originalmente propostas pelos artigos referenciados, serão referidos em inglês. A não tradução destes termos, que teriam de ser directamente traduzidos com expressões mais comuns utilizadas na língua portuguesa (eventualmente menos precisas), visam facilitar a leitura, a rapidez do entendimento do leitor, as posteriores consultas mais detalhadas aos trabalhos referidos, ou o eventual estabelecer de relacionamentos com outros temas.

Por exemplo, a utilização das expressões *area, unit, topic* serão facilmente identificadas a determinados assuntos e aspectos de alguns dos artigos referidos, enquanto que a sua tradução directa para área, unidade ou tópico, traria mais dificuldades de interpretação, precisão e dimensão na escrita.

Capítulo 1
Introdução

Neste capítulo é feita uma descrição introdutória da dissertação, contextualizando a motivação que lhe deu origem, uma apresentação resumida da problemática que aborda, bem como uma definição do âmbito pretendido dentro da problemática.

O começo é um momento sempre delicado, pelo que, neste capítulo introdutório da dissertação estão presentes diversos objectivos, o enquadrar e apresentar da motivação que lhe deu origem, o descrever do problema que se pretende abordar e, considerando o mesmo, o detalhar do âmbito e objectivos concretos do trabalho e da solução proposta. Por fim, na última secção do capítulo, apresenta-se a estrutura deste documento, detalhando os temas dos capítulos seguintes.

1.1 Enquadramento e motivação

Nesta secção, como forma de introdução ao tema do trabalho da dissertação, apresenta-se o âmbito onde se enquadra, bem como a motivação para o seu desenvolvimento.

1.1.1 O Processo de Bolonha

A "Declaração de Bolonha" [1] [2], que toma o nome da cidade onde foi assinada em 1999, pretende estabelecer o guia para a criação de um espaço de europeu de ensino superior até 2010.

O chamado "Processo de Bolonha" [3] [4] [5], que envolveu no seu desenvolvimento diversas iterações de 1999 até 2007 [5], entende-se como a implementação da declaração, seguindo os seus principais objectivos:

i. A criação no âmbito do espaço europeu, um sistema de graus académicos, comparável e facilmente compreensível por todos;
ii. A adopção de um sistema baseado em 3 ciclos de ensino (1º ciclo - Licenciatura, 2º ciclo – Mestrado e 3º ciclo - Doutoramento) [6];
iii. A promoção da mobilidade, quer para estudantes, quer para professores, investigadores, etc.;
iv. A promoção da cooperação europeia na certificação da qualidade do ensino superior, para obtenção de critérios e metodologias comparáveis;
v. A promoção e dinamização das sinergias entre as áreas de ensino e investigação europeias.

Um instrumento fundamental para permitir a construção de um sistema de graus académicos que seja comparável ao nível europeu e que facilite a mobilidade dentro do espaço de ensino, é o sistema ECTS – *European Credit Transfer and Accumulation System* [7]. Este permite, na sua essência, definir uma medida do esforço que um aluno terá para a obtenção de determinado conjunto de competências. Como o nome indica, o sistema pretende obter uma forma de acumulação e transferência de créditos, orientando a comparação e compatibilização ao nível europeu do esforço de aluno [8].

Esta medida de ECTS (ou créditos ECTS) está directamente relacionada com a carga de trabalho do aluno, habitualmente envolvida para o alcançar com sucesso dos resultados e objectivos de aprendizagem das várias actividades que deverá desenvolver [9]. Esta ser carga ou esforço poderá ser directamente medida em **horas de trabalho** e aplica-se a todas as actividades de aprendizagem que envolvem esforço do aluno, sejam aulas teóricas ou práticas, seminários, projectos, trabalhos práticos, auto-estudo, exames, etc.

Como valor base, o sistema define que **60 créditos ECTS** corresponderão habitualmente à carga de trabalho de um ano académico a tempo inteiro, bem como aos respectivos resultados de aprendizagem pretendidos. O sistema considera que, em geral, o esforço de um aluno varia entre 1500 e 1800 horas por ano e que, correspondentemente, cada crédito ECTS terá um valor entre 25 e 30 horas de trabalho [10].

Naturalmente, os créditos ECTS serão atribuídos às várias componentes educacionais existentes nas universidades, sejam disciplinas, trabalhos de dissertação ou laboratoriais, etc., estando directamente indexados à carga envolvida de trabalho do aluno. Por exemplo, as disciplinas nas universidades poderão habitualmente variar entre **3 e 8 ECTS**, sendo esse valor directamente atribuído ao aluno, caso tenha sucesso no alcançar de resultados nessa disciplina (directamente relacionado com a componente de avaliação da mesma).

Como exemplos, e considerando a Universidade Nova de Lisboa [11] [12], um semestre da licenciatura em Eng.ª Informática (o 1º ciclo) corresponderá geralmente a 30 ECTS, a licenciatura (os 3 anos) a um mínimo de 180 ECTS, o mestrado em Eng.ª Informática (Bolonha, 2º ciclo, 2 anos), a um mínimo de 120 ECTS e relativamente ao doutoramento em ciência da computação (3º ciclo, 3 anos), a parte escolar corresponde a um mínimo de 30 ECTS, sendo naturalmente seguida da respectiva tese. Para efeitos de comparação, o valor definido pela Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (e aplicado transversalmente aos seus cursos) de correspondência entre 1 ECTS e esforço do aluno são **28 horas** [13] [14], o que significa que o semestre tipo referido acima corresponderá a 840 horas de esforço do aluno.

Este sistema de ECTS, aliado a instituições que garantam a execução de sistemas de certificação e controlo de qualidade, são factores que contribuem para o cumprimento dos objectivos da declaração de Bolonha, mais concretamente, dois dos seus principais pilares, a possibilidade de **comparabilidade** real entre os diferentes programas, cursos e universidades, bem como a simplificação dos processos de **mobilidade** dos alunos entre os mesmos. Efectivamente, estes factores são críticos para a criação, manutenção e promoção de um espaço europeu comum de ensino superior.

Adicionalmente, considerando que o sistema de ECTS está directamente relacionado com o esforço necessário para um aluno médio atingir determinado conjunto de competências, no processo de Bolonha foi também efectuada uma revisão deste conjunto principal de competências, incluído não só competências

técnicas e científicas, mas também estendendo as mesmas a competências de comunicação, honestidade, humanas e de personalidade, hábitos pessoais, etc., vulgarmente conhecidas como *soft-skills* [15] [16]. Este aspecto será abordado mais em detalhe mais à frente no documento.

Por fim, é importante reter que o processo de Bolonha não aborda especificamente problemáticas de formas de ensino, compatibilização de temas dentro das distintas áreas ou da cobertura dos cursos ao nível temático nas respectivas áreas de conhecimento. Como se poderá ver mais à frente no caso específico da área da ciência da computação, estes são aspectos específicos de cada área.

1.1.2 O processo de Bolonha em Portugal

Com uma participação directa no processo de Bolonha [17] desde do início, o governo português iniciou em 2005 medidas de adequação ao processo, tendo aprovado legislação neste sentido entre 2005 e 2006 [18], respectivamente: a Lei nº 49/2005, de 30 de Agosto [19], que estabelece as bases para a adopção do modelo de organização do ensino superior em 3 ciclos, para a adopção do sistema de créditos europeu (ECTS) e para a transição para um sistema de ensino baseado no desenvolvimento de competências; e o Decreto-Lei nº 74/2006, de 24 de Março [20], que estabelece a regulamentação para as alterações da lei de bases acima, i.e., para a adopção dos principais pontos do Processo de Bolonha (o modelo de 3 ciclos e o sistema de créditos) e respectivas formas de transição do anterior formato.

Tendo Portugal assumido a presidência do processo de Bolonha na 2ª metade de 2007 [3], este suporte legal acima descrito (e ao qual foi adicionado, como suporte aos regimes de transição, o Decreto-Lei nº 107/2008 de 25 de Junho de 2008 [21]), aliado ao respectivo esforço de conversão dentro das universidade portuguesas, garantiu o cumprimento das datas impostas para a adopção generalizada na Europa deste modelo até 2010 [21]. O longo período de tempo e a necessidade de adaptação relativamente urgente a Bolonha, originaram diversas vagas de adopção do modelo nas universidade portuguesas, uma primeira vaga para o ano lectivo de 2006/2007 (da qual o curso de Eng.ª. Informática da FCT-UNL fez parte) com uma adopção de cerca de 38% dos cursos, uma segunda vaga no ano lectivo de 2007/2008 com a adopção de cerca de 90% dos cursos e finalmente, uma terceira vaga com os restantes cursos no ano lectivo de 2008/2009 [22].

Estes processos de remodelação a Bolonha, de forma a garantirem a sua acreditação, seguiram essencialmente um conjunto de elementos burocráticos, consolidados sob a forma documental num relatório, que os cursos e respectivas universidades entregaram ao ministério da educação. Estes elementos estão especificados no artigo 63º do Decreto-Lei nº 74/2006 [20] e são essencialmente documentos e não modelos eventualmente computáveis:

a) *"A indicação dos ciclos de estudos em funcionamento que são objecto da adequação"*;
b) *"Os objectivos visados pelo ciclo de estudos"*;
c) *"A fundamentação do número de créditos que, com base no trabalho estimado dos alunos, é atribuído a cada unidade curricular, incluindo os inquéritos realizados aos estudantes e docentes tendo em vista esse fim"*;
d) *"A fundamentação do número total de créditos e da consequente duração do ciclo de estudos"*;
e) *"A demonstração da adequação da organização do ciclo de estudos e metodologias de ensino [...] à aquisição de competências"*;
f) *"Uma análise comparativa entre a organização fixada para o ciclo de estudos e a de cursos de referência com objectivos similares ministrados no espaço europeu"*;
g) *"A forma como os resultados da avaliação externa foram incorporados na organização do ciclo de estudos"*.

Inclusive na união europeia, as recomendações relacionadas com o processo de Bolonha e respectiva *Framework*, são naturalmente orientados a elementos documentais, como se pode verificar nos documentos de suporte do site oficial [3].

1.1.3 Motivação

Como foi possível verificar, o processo de Bolonha é efectivamente algo de extrema importância ao nível europeu, dando especial ênfase às questões de mobilidade dos alunos e comparabilidade entre os cursos. No entanto, o processo e sua acreditação nos vários países, baseia-se na *Framework* de Bolonha e na normalização de regras (a adopção de ECTS, por exemplo), que como se verificou são essencialmente estabelecidos sobre a forma documental.

Estes modelos sob a forma documental, i.e. papel e documentos, não são o ideal para resolver eficazmente, por exemplo, a questão da comparabilidade. A eventual introdução de um modelo computável, compatível ao processo de Bolonha, permitiria algumas aplicações interessantes, tais como, por exemplo:

- Sistemas de apoio à elaboração de planos de curso para alunos de transferência. Teriam o objectivo de facilitar, eventualmente de forma assistida, o planeamento de cursos para alunos de transferência (entre universidades e / ou cursos da mesma universidade), tendo presente a proveniência e actual curriculum do aluno ao nível das suas competências e anteriores disciplinas frequentadas, bem com a respectiva cobertura do mesmo perante o novo curso.

- Integração com sistemas de gestão de informação bibliográfica, para recomendação / pesquisa de bibliografia adequada para as disciplinas (ou parte das disciplinas). Considere-se que poderá existir detalhe nessa bibliografia relativo ao seu impacto nas competências. Tal integração e detalhe poderá servir quer a alunos, quer a professores, permitindo eventuais cruzamento de bibliografias entre disciplinas, não apenas nas focado nos temas a abordar, mas também nas competências a obter.

- A possível criação de um sistema europeu de apoio / suporte aos alunos *Erasmo*, permitindo uma comparabilidade entre as disciplinas das várias universidades, quer ao nível do impacto nos planos curriculares originais dos alunos, quer o nível do impacto nas competências finais a obter pelos mesmos. Este possível sistema (ou sistemas) introduziria várias funcionalidades, tais como, por exemplo, a pesquisa de universidades / escolas que lhe oferecem disciplinas para completar a sua formação fora da sua universidade de origem, a capacidade de perceber antecipadamente qual o impacto que frequentar determinada disciplina fora da sua universidade de origem irá ter no seu actual curso e competências, etc.

- A possibilidade dos empregadores terem acesso às estruturas curriculares e competências obtidas genericamente pelos alunos que frequentam os vários cursos (sejam licenciaturas e mestrados), mas também a possibilidade de recrutarem perfis conforme uma especificação dos mesmos orientada às competências. Isto permitiria aos empregadores identificarem quais os cursos que melhor preparam alunos para o mercado de trabalho, com as competências que mais se adequam às suas necessidades.

- Considerando, por fim, que alunos não são pessoas iguais e que os seus os percursos curriculares são habitualmente diferentes, devido à existência de disciplinas opcionais e eventuais ramos dentro dos cursos, bem como as naturais diferenças dos resultados das avaliações, cada aluno terá um efectivo perfil distinto, com as respectivas variações das suas competências. Considerando o cenário de um modelo computável, era muito interessante ter um perfil completo e detalhado das **competências** do aluno ao longo do seu percurso na universidade, o que poderia ser utilizado para a optimização das escolhas curriculares do aluno (numa perspectiva de orientação às competências) bem como na perspectiva da empregabilidade (acima referida).

A existência de um modelo computável introduziria também alguns benefícios directos aos processos de acreditação, que além deste processo inicial de transição a Bolonha, deverão também ter presente as questões de manutenção da qualidade e da creditação. Um modelo computável seria naturalmente constituído por definições claras e reutilizáveis dos descritivos das competências, dos planos e percursos curriculares, etc., que poderiam servir de base para simplificar e assegurar os aspectos de manutenção, qualidade e evolução das creditações de Bolonha.

Um outro aspecto relevante, que poderá estar associado à criação de modelos computáveis e de certa forma relacionado com o parágrafo anterior, é a eventual necessidade da existência de *standards* para a troca de informação. Por exemplo: a existência de certificados digitais dos alunos, contendo todas as informações que justificariam os ECTS por ele adquiridos; a especificação digital da oferta curricular de uma instituição, etc.

Os vários exemplos anteriormente referidos podem ser interpretados como futurologia (possível e desejável), mas são apresentados para evidenciar a necessidade, ou pelo menos as possibilidades, de um modelo computável para a representação dos cursos.

1.2 Apresentação do problema

Tendo presente as aplicações sugeridas na secção anterior, é importante perceber que são de certa forma futurologia, mas ao mesmo tempo aspectos de difícil concretização. Modelos possíveis de serem globalmente aplicáveis, que é o caso referido atrás, envolvem diversos esquemas de captura de conhecimento científico do ensino superior, implicando um acordo e normalização em todas as instituições de ensino, bem como acesso à informação dos vários serviços de suporte (secretarias, etc.) e dos sistemas de informação já existentes, o que torna bastante complexa a tarefa de concretização destes modelos.

Na próxima secção 1.2.1, apresenta-se uma panorâmica geral e informal da forma de definição típica dos cursos, detalhando os actores e principais processos envolvidos, na perspectiva deste contexto de Bolonha e numa eventual aproximação às visões futuristas anteriormente descritas.

Na secção posterior 1.2.2, será abordado o contexto específico desta dissertação – a área da ciência da computação, bem como alguns esforços desenvolvidos dessa mesma área.

1.2.1 A definição de cursos – uma panorâmica geral e processos

Na criação ou definição de um conjunto de cursos numa universidade, podem ser considerados conceptualmente um conjunto de fases ou processos:

i. A normalização e conceitos gerais;

ii. A especificação das áreas de conhecimento;
iii. A estruturação curricular (cursos e respectivas disciplinas);
iv. A definição dos métodos de ensino na disciplina;
v. A definição dos métodos de avaliação na disciplina.

Considerando estes processos na perspectiva do controlo de qualidade do ensino, cada um deles poderá ser um ponto importante quer na avaliação da qualidade, quer para hipóteses de introdução de melhorias, podendo globalmente ser considerados como fazendo parte de um processo transversal de melhoria contínua, cíclico e com várias iterações, iniciado tipicamente pela 1ª fase, a especificação de conceitos e normalizações a utilizar no conjunto de cursos.

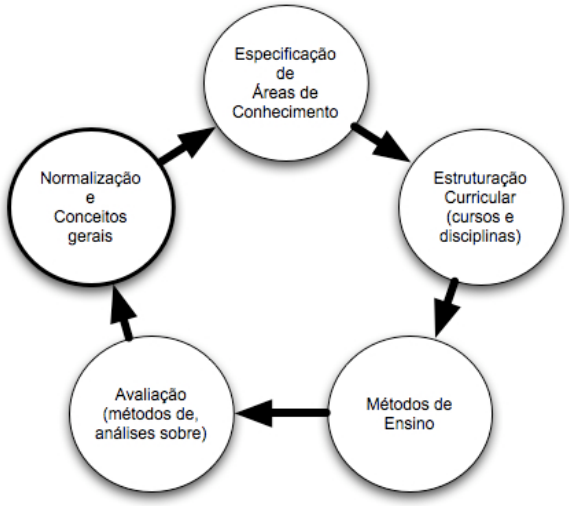


Figura 1.1 – Processo cíclico de melhoria contínua no ensino

A Figura 1.1 representa uma visão global do fluxo típico de definição de cursos, considerando aspectos de melhoria contínua, com uma avaliação permanente em cada processo, onde as experiências e aprendizagens são passadas sucessivamente para o processo seguinte. Numa análise mais detalhada de cada processo:

- **Normalização e conceitos gerais**
  Como forma de possibilitar uma "linguagem comum" em toda a universidade, facilitando assim a partilha de informação, a partilha de disciplinas comuns e a comparação entre os esforços envolvidos nas disciplinas e respectivas avaliações, é necessário efectuar processos de especificação e clarificação de conceitos ao nível da universidade, normalizando critérios, como por exemplo: qual o significado utilizado para 1 ECTS e quantas horas são realmente estimadas, 26h? 28h?; A adopção pela universidade do conjunto típicos de valores para os ECTS das disciplinas, considerando as normas exteriores relativas aos ECTS

totais para uma licenciatura, um mestrado, etc.; Quais os objectivos gerais para uma licenciatura, um mestrado e um doutoramento, respectivamente a aquisição de competências, a especialização e a inovação;

Este processo é complicado de executar e promover regularmente, pelo facto de ser só realmente útil num nível global na universidade.

- **Especificação das áreas de conhecimento**

  Como forma de melhor definir os cursos dentro das universidades, é importante que sejam bem definidas quais as temáticas a abordar dentro das áreas dos cursos, facilitando assim uma melhor divisão (entre os cursos) e uma correcta formação dos alunos. Isto permitirá ainda particionar tematicamente as disciplinas dos cursos, possivelmente facilitando algum cruzamento temático (via disciplinas) entre os mesmos. Esta especificação, conceptualmente, deverá ser executada por uma comissão de especialistas dentro da área.

- **Estruturação curricular (cursos e disciplinas)**

  Este processo pode ser decomposto em 2 partes, a definição da estrutura do curso (onde se definem quais as disciplinas e sua distribuição nos anos / semestres) e a definição das disciplinas em concreto (quais os tópicos a abordar, a sua estruturação, quais as competências a adquirir, as avaliações, etc.). Apesar dos processos serem representados de forma agrupada por uma questão de simplificação, estes são executados com objectivos distintos e provavelmente por diferentes responsáveis.

  A definição da estrutura curricular do curso deverá ser abordada por um grupo de responsáveis do mesmo, por exemplo, uma comissão científica. Esta definição terá como motivação a melhor organização ao nível da aprendizagem, para o assegurar das principais competências a adquirir pelo aluno, cobrindo as áreas temáticas definidas para o curso.

  A definição das disciplinas (e subsequentes competências a adquirir nas mesmas), será da responsabilidade de, ou do mesmo grupo responsável pelo curso, ou mais provavelmente, por um grupo responsável pela área temática da disciplina e que será responsável também por definir as várias disciplinas incluídas nessa área.

- **A definição dos métodos de ensino**

  Também de certa forma integrado na definição das disciplinas, é importante que seja abordada a problemática dos métodos de ensino. O objectivo será identificar as melhores formas de abordar a temática das disciplinas numa perspectiva de Bolonha, ou seja, não apenas numa visão de optimização da transmissão do conhecimento

programático, mas acima de tudo, a promoção da melhoria da aquisição de competências (técnicas ou outras) pelo aluno.
Esta problemática poderá ser abordada a vários níveis, transversalmente na universidade (e portanto aplicável aos vários cursos), apenas ao curso, ou independentemente à disciplina. Conforme o nível, podem existir diversos responsáveis, seja uma comissão na universidade ou o regente responsável pela disciplina. O processo, apesar de poder ter variados âmbitos, é importante para a qualidade pois influencia directamente a aquisição de competências pelo aluno, sejam estas técnicas ou outras.

- **A definição da avaliação (métodos de e análise sobre)**
  Por fim, no fluxo de definição (e redefinição) dos cursos, o processo de avaliação é igualmente importante para a qualidade, pois tem também influencia directa na verificação do estado de aquisição de competências pelo aluno.
  Este processo não passa pela execução da avaliação, mas sim pela definição ou revisão dos métodos utilizados na avaliação, bem como a posterior análise das avaliações, o que poderá ser um instrumento muito útil do controlo de qualidade.
  Como o processo anterior, o âmbito deste poderá variar entre a universidade e a disciplina, sendo os responsáveis igualmente comissões transversais à universidade ou regentes da disciplina. O conhecimento recolhido deste processo, aliado a todos os outros, é importante para possíveis revisões dos cursos e disciplinas, pois permite, novamente numa perspectiva de Bolonha, verificar e certificar a forma como as competências foram ou estão a ser adquiridas pelos alunos.

Estes processos não são triviais, pois envolvem diferentes níveis dentro das universidades e distintas responsabilidades na sua execução / definição. Se pensarmos nestes processos num visão diferente, *top-down*, em sequência, com os âmbitos e actores responsáveis descritos acima, bem como alguns dos processos mais detalhados, é possível produzir-se uma representação na Figura 1.2.

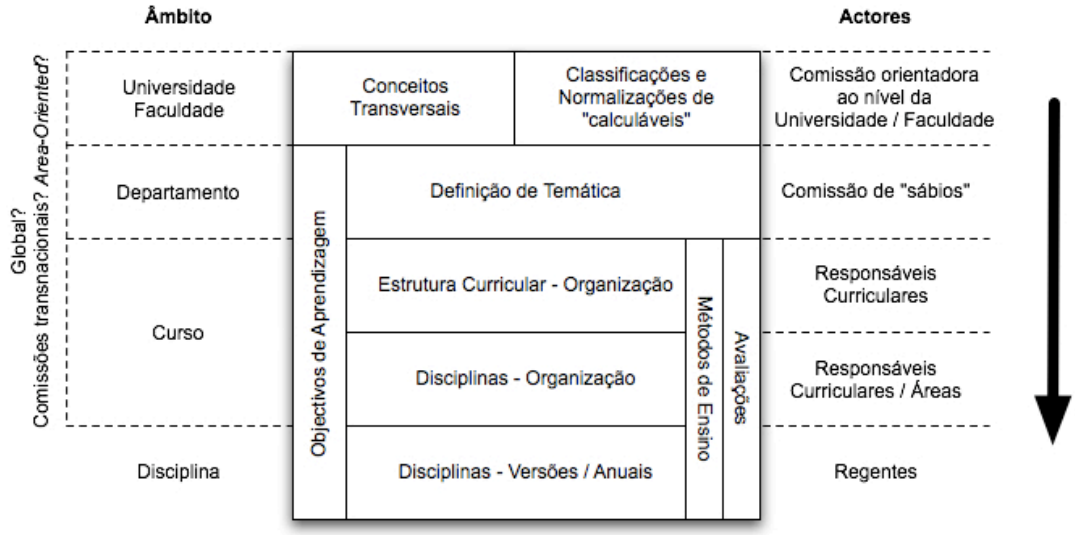


Figura 1.2 – Detalhe dos processos envolvidos na definição curricular de cursos e disciplinas

As principais alterações à figura inicial estão na decomposição do processo de definição do curso em 3 partes, o processo da organização da estrutura curricular, o processo de definição / organização de uma disciplina e a gestão / reorganização da disciplina nas suas diferentes versões anuais. Estes 2 últimos diferem relativamente aos objectivos concretos, enquanto que no primeiro o objectivo é focar nos aspectos da organização dos tópicos e temas a abordar, no segundo focam-se mais aspectos relacionados com a gestão típica da disciplina anual, como os métodos de avaliação e organização dos horários, etc.

Adicionalmente, alguns processos estão representados verticalmente de forma a expressar que são aspectos que podem "atravessar" ou estar envolvidos em vários processos e nos seus respectivos âmbitos / actores. Os processos relacionados com a avaliação e os métodos de ensino, inserem-se nesta categoria, podendo ainda ser considerandos dentro das normalizações e conceitos transversais na universidade (os processos de topo).

Um outro aspecto importante na Figura 1.2 é o acrescento de um outro processo vertical com o nome **Objectivos de aprendizagem**. Este é, como o nome indica, o processo representativo da definição dos objectivos de aprendizagem das disciplinas e dos cursos. No entanto, por estar representado na vertical e ao longo dos vários níveis, significa que poderá ter um âmbito de execução também mais ao nível transversal na universidade (além do seu âmbito no curso).

Esta representação visual dos vários processos permite ter uma noção da complexidade do tema, do seu âmbito alargado e do facto de serem processos executados ao longo de diversas fases do ciclo académico.

Por fim, para que estes processos não sejam unicamente burocráticos, i.e., suportados sob a forma simplesmente documental e requerendo, por exemplo, a intervenção de especialistas para avaliação dos currículos, é importante considerar-se um modelo computável. Para se atingir este objectivo, é importante

centrar-se primeiramente na procura de modelos base para a definição do conteúdo curricular das disciplinas, dos objectivos de aprendizagem, etc. Assim, na secção seguinte, e relacionado com a procura destes modelos, será apresentado o esforço desenvolvido pelo ACM e IEEE, relativo à área da ciência da computação.

1.2.2 Na área da ciência da computação

Em concreto, na ciência da computação (em inglês *computer science* - CS) [23], que é o enquadramento da aplicação desta dissertação, e talvez por ser uma área mais recente, existem já algumas tentativas de estruturação dos conceitos base e formas dos processos de ensino [24].

A *Association for Computing Machinery* (ACM)[1], sendo a primeira associação mundial da área científica e educacional da computação e a *Institute of Electrical and Electronics Engineers* (IEEE)[2] *Computer Society* (IEEE-CS)[3], sendo uma unidade organizacional dentro da IEEE focada na área da computação, têm efectuado diversas abordagens à partilha de informação, à identificação e consolidação das áreas de conhecimento da ciência da computação, bem como à identificação de regras e boas práticas acerca dos processos de ensino e estruturas dos currículos da área.

Em concreto, é possível referir alguns trabalhos importantes:

- ***ACM Computing Classification System* (ACM-CCS)**
  Um sistema de classificação de tópicos para a área CS, ou seja, uma abordagem a uma taxonomia primeiramente orientada à classificação e identificação de tópicos sobre trabalhos científicos [25] [26] [27].

- ***ACM/IEEE-CS Computer Science Body of Knowledge* (CS-BoK)**
  Um abordagem à organização temática das áreas de conhecimento dentro da ciência da computação [30], englobado no trabalho referido no próximo ponto [28], que pode servir de base para a identificação de temas específicos para o ensino.

- ***ACM/IEEE-CS Computing Curricula* 2001 (CC2001)**
  Um conjunto de recomendações para cursos na área CS [28] [29], focadas mais em concreto no 1º ciclo de Bolonha, onde além de serem estipuladas bases para um CS-BoK (acima), são propostos modelos para as estruturas curriculares ou planos de cursos, são propostas as temáticas mínimas e obrigatórias para as estruturas (referidas como

[1] http://www.acm.org/

[2] http://www.ieee.org/

[3] http://www.computer.org/

disciplinas *core*), são propostos conjuntos de objectivos de aprendizagem a cada temática (destinados a promover de forma objectiva a avaliação do sucesso do aluno, que se relaciona com a aquisição de competências) e por fim, propostas detalhadas de várias disciplinas (1º ciclo e algumas avançadas, possíveis de serem englobadas no 2º ciclo).

- **ACM/IEEE-CS/AIS Computing Curricula 2005 (CC2005)**
  Sendo natural a necessidade de evolução dentro duma área recente como é a CS, o grupo de trabalho envolvido no CC2001, com a inclusão adicional da *Association for Information Systems* (AIS)[4], percebeu que não seria viável produzir um único trabalho para a actualização das disciplinas bem como introduzir a totalidade de novas disciplinas e temáticas, pelo que foi decidido detalhar os currículos por subárea dentro da CS (por exemplo, tecnologia da informação, engenharia de *software*, *etc*.) e produzir um relatório de introdução de nome CC2005 *Overview Report* [31], focando como anteriormente no 1º ciclo. Este relatório aborda as características de cada subárea, as suas similaridades e diferenças, bem como as respectivas características dos alunos que as frequentam.
  As subáreas identificadas são apresentadas na Figura 1.3, tendo sido produzida para cada uma o seu respectivo relatório (identificado pelo nome da subárea e ano de produção do mesmo), onde são detalhadas as subáreas de conhecimento e temáticas próprias, disciplinas específicas e respectivas abordagens à estrutura curricular.

Estes trabalhos aqui referidos são peças importantes na normalização e estruturação da educação da área CS, servindo de base e inspiração para esforços no sentido de produção de um eventual modelo computável. Naturalmente, estes trabalhos já terão de certa forma inerente a perspectiva do processo de Bolonha, pelo que se focam numa orientação às competências e objectivos de aprendizagem.

Por fim, a área da CS está também relacionada com outras áreas de conhecimento, como a Matemática, Física, Electrónica, etc., principalmente na perspectiva do ensino, como forma de estabelecer uma base comum (*background*) de conhecimento científico.

[4] http://www.aisnet.org/

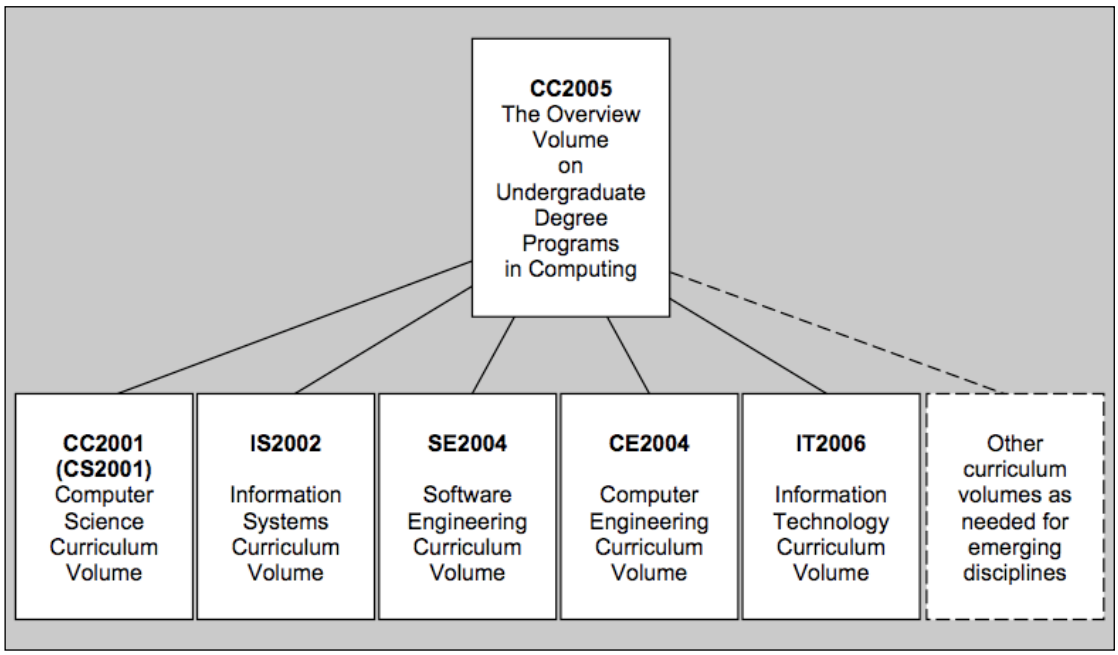


Figura 1.3 - Estrutura da *Computing Curricula Series*[5]

Após alguma pesquisa, não foi encontrado nas áreas da Matemática e da Física, trabalho substancial no sentido um *body of knowledge*[6] consensual, como se viu anteriormente para a área de CS. Foi, no entanto, encontrado na área da Matemática, algum trabalho prévio sobre uma possível classificação taxonómica, desenvolvido pela *American Mathematical Society* (AMS), com o nome "*Mathematics Subject Classification*" em 2000 [32] e uma revisão recente em 2010 [33] (ainda em validação), mas que não poderá ser considerado com uma recolha exaustiva da área de conhecimento. Para se pensar de forma abrangente no apoio à controlo de qualidade do ensino na área de CS, será necessário abordar a problemática da representação de conhecimento das disciplinas destas áreas e a sua possível comparabilidade (e computabilidade) com as disciplinas CS.

Assim, à luz desta secção, a principal motivação desta dissertação estará focada na área da ciência de computação, sendo apresentada detalhadamente na secção seguinte.

1.3 Âmbito e objectivos da tese

Esta dissertação tem como objectivo abordar o problema descrito na secção anterior, mas restringindo o seu foco à problemática da especificação de um

[5] Fonte: CC2005, pág. 7

[6] *Body of Knowledge (BoK) - is a term used to represent the complete set of concepts, terms and activities that make up a professional domain, as defined by the relevant professional association (Source: WikiPedia).*

modelo de informação computável [34] [35] [36], que permita uma abordagem aos processos definição de um curso, tendo presente a perspectiva do processo de Bolonha.

Restringir o âmbito do modelo a um curso, significa que se pretende propor soluções para os processos ao nível do curso, conforme apresentados na secção anterior. Ou seja, conforme a seguinte Figura 1.4, a definição da temática e os processos de definição da estrutura curricular, que incluem as disciplinas, os métodos de avaliação e ensino, os objectivos de aprendizagem e competências, etc., prevendo sobre os mesmos, aspectos de manutenção, evolução e controlo de qualidade.

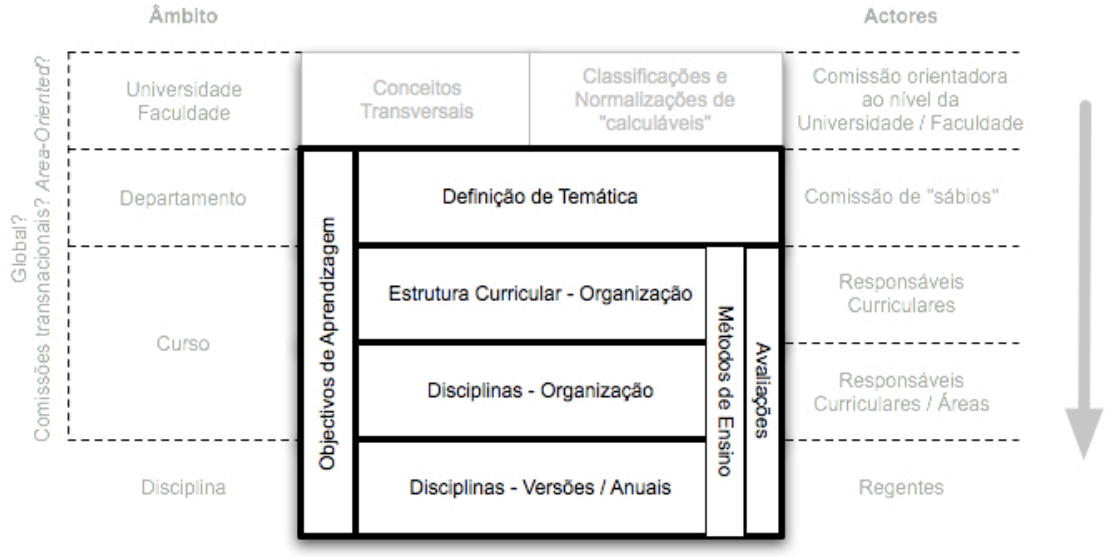


Figura 1.4 – Identificação dos processos "alvo", âmbito da dissertação

Propõe-se um trabalho com o principal objectivo focado maioritariamente na especificação do modelo, respeitando as seguintes características:

- **Um modelo** que se considere **"aberto"**. Isto é, que:
  - Tenha presente a necessidade de eventual aplicabilidade a vários **tipos de cursos**;
  - Permita suporte à **informação sobre várias áreas de conhecimento**, considerando aspectos de **normalização** das mesmas para eventual partilha e **disponibilização ao exterior**;
- Garantir a representação da **informação sobre a forma computável** [36], de forma a que esta possa ser facilmente manipulada, cruzada e

aplicável em eventuais instrumentos de análise (relaciona-se com a motivação da dissertação, conforme secção anterior);

- Tenha presente as necessidades de **extensibilidade** e **evolução incremental**, de forma a lidar com as sucessivas iterações do processo de ensino, mantendo um funcionamento coerente;

- Permita a recolha e organização das **temáticas** a abordar e dos **conceitos para definição e organização das disciplinas**, incluindo métodos de avaliação, bibliografia, etc.;

- Permita a **organização e estruturação do plano curricular** do curso;

- Ter presente as principais **considerações de Bolonha**, em particular a orientação do ensino ao paradigma de **aquisição de competências**;

Pretende-se também nesta dissertação, e como objectivos mais secundários, a definição de uma **possível arquitectura modular** e **implementação de um protótipo**, para efeitos ilustrativos de alguns dos aspectos e características do modelo. Pretende-se com isto demonstrar a capacidade computável do modelo, a análise de alguns aspectos de normalização, aliado à produção de alguns exemplos de resultados ou instrumentos de análise possíveis, descrevendo em simultâneo as dificuldades que se antevêem na aplicabilidade e construção de uma solução adequada ao "mundo real".

O âmbito de dissertação (incluindo-se modelo e protótipo) é delimitado à **área da ciência da computação**, em particular a **Licenciatura de Eng.ª Informática** e ao **1º ciclo**, pois definiu-se como base inicial a utilização dos trabalhos ACM / IEEE, referidos anteriormente, respeitantes à normalização das temáticas e estruturação curricular. Em concreto, o trabalho CC2001 [28] [29] é de especial importância para o desenvolvimento nesta dissertação.

Utilizar-se-á o nome "MODELO PARA APOIO À DEFINIÇÃO CURRICULAR DE CURSOS 1º CICLO NA ÁREA DA CIÊNCIA DA COMPUTAÇÃO" (MADCC-CC) para referir o protótipo exemplificativo e respectiva arquitectura base proposta.

Com esta abordagem, i.e., a especificação de um modelo computável, a criação de um pequeno protótipo e definição de uma possível arquitectura base correspondente, efectua-se uma prova de conceito, aplicado unicamente à área da ciência da computação, onde se pretende avaliar que a definição e normalização de conceitos do ensino, aliado à recolha de informação sobre a mesma base, permite produzir instrumentos úteis para a especificação, apoio à especificação e de análise sobre a completude, cobertura e qualidade dos cursos do ensino universitário.

1.4 Estrutura do documento

Este documento está estruturado pelos seguintes capítulos: o **capítulo 2**, onde se efectua uma descrição e resumo de alguns trabalhos que se relacionam com o tema da dissertação; o **capítulo 3**, onde se efectua uma descrição geral do modelo proposto, onde são detalhados em maior pormenor os conceitos principais e se elabora sobre o povoamento, extensibilidade, e exploração eventual do modelo; o **capítulo 4**, onde se apresenta uma proposta de arquitectura para eventual solução de implementação do modelo e onde se descreve a implementação de um pequeno protótipo da solução e tecnologia utilizada; o **capítulo 5**, onde se efectua uma avaliação do trabalho efectuado, indicação do possível trabalho futuro sobre a arquitectura, bem como conclusões finais; Por fim, os anexos a este documento estarão sobre a forma digital.

Capítulo 2
Trabalhos relacionados

Neste capítulo é feita uma descrição dos principais trabalhos relacionados com a temática desta dissertação, bem como um estado da arte sobre os tópicos e tecnologias que se abordaram ao longo do trabalho efectuado.

Neste capítulo apresentam-se resumidamente os principais trabalhos, metodologias e tecnologias, que estão relacionados com o tema da mesma ou que, de alguma forma, foram relevantes para o trabalho prático efectuado. Abordam-se principalmente trabalhos que especifiquem e normalizem formas de recolher a informação da área de conhecimento CS, bem como trabalhos que proponham modelos relacionados com o ensino, seja para definição curricular, definição de competências, etc. Note-se, no entanto, que outras tecnologias e trabalhos relacionados directamente com o desenvolvimento do protótipo e tecnologias associadas, estarão descritos no capítulo respeitante ao mesmo, o Capítulo 4.

. Os principais trabalhos desenvolvidos pela ACM e pela IEEE (entenda-se, principais no âmbito desta dissertação), já previamente referidos, são de especial importância para o trabalho aqui desenvolvido, pois estabelecem normalizações e definições, quer ao nível científico, quer ao nível do ensino, que devem ser encaradas como recomendações, guias ou boas práticas [31].

Em concreto, na secção 2.1 aborda-se um sistema de classificação de tópicos proposto pela ACM (ACM-CSS). Na secção 2.2, apresentam-se com algum detalhe os guias ACM de cobertura temática e de proposta de estruturação curricular, CC2001 e CC2005, referindo-se na secção 2.3, diversos trabalhos especificamente focados no modelação dos conceitos destes guias. Na secção 2.4 abordam-se trabalhos referentes ao processo de Bolonha e o seu maior foco no conceito das competências, como os descritores de Dublin e a taxonomia de *Bloom*, sendo na secção 2.5 descrito mais em particular o trabalho e esforço desenvolvido no D.I. para adequação do curso ao processo de Bolonha. Por fim, na secção 2.6, elaboram-se algumas conclusões e considerações à aplicabilidade nesta dissertação dos trabalhos apresentados.

2.1 ACM Computing Classification System (ACM-CSS)

Este trabalho, como o nome indica, é um sistema de classificação de tópicos e de indexação, que tem sido aplicado maioritariamente à literatura publicada na área da computação (livros, artigos, etc.) [25] [26] e que estabelece uma primeira abordagem a uma taxonomia.

Este sistema de classificação, originalmente chamado de *Computing Reviews Classification System* – CRCS, tem sido aplicado ao longo dos últimos 30 anos principalmente para a classificação de documentos dentro desta área (por exemplo, na classificação de todos os documentos no "*ACM Guide to Computing Literature*" [26]). Sofreu também diversas revisões durante o seu ciclo de vida, como forma de acomodar a natureza fortemente evolutiva da área [26], sendo a última versão de 1998 (referido com CCS98) [27] [37].

O actual sistema consiste essencialmente [37] em definição de uma estrutura hierárquica em árvore com 4 níveis, dos quais os 3 primeiros são geralmente níveis com códigos específicos (*coded levels*) do tipo letras e números, enquanto

o último nível contém descrições específicas de assuntos sem códigos (*subject descriptors*). Em cada um dos níveis codificados existem nós (*nodes*) que contém, além do código, breves descrições (algumas palavras) identificando aquilo que se pretende categorizar. O 1º nível contém 11 nós, identificados por letras de A a K, tendo geralmente 1 ou 2 níveis filhos, identificados por letras e números, que começam sempre (exceptuando se forem o último nível) com um nó do tipo *General* (o nó "0") e terminam com um nó do tipo *Miscellaneous* (o nó "m"). O último nível, os tais *subject descriptors*, são considerados como as folhas da árvore. A última versão da estrutura hierárquica do CSS98 existe disponível no site ACM [26] sobre vários formatos, como HTML, ASCII e XML.

Apresenta-se em seguida a parte inicial da árvore de classificação [27], como forma de exemplificar melhor o descrito no parágrafo anterior:

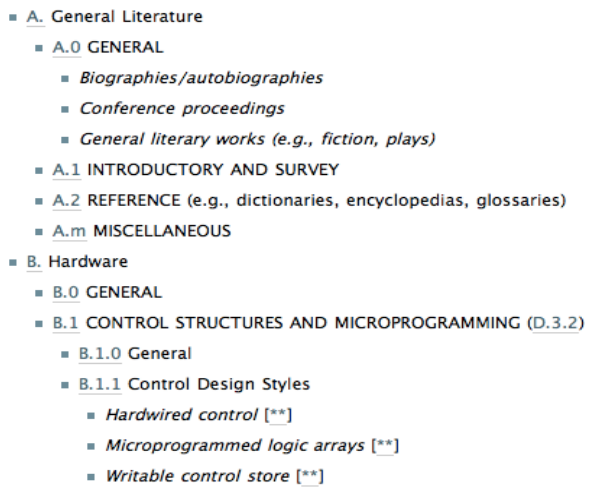
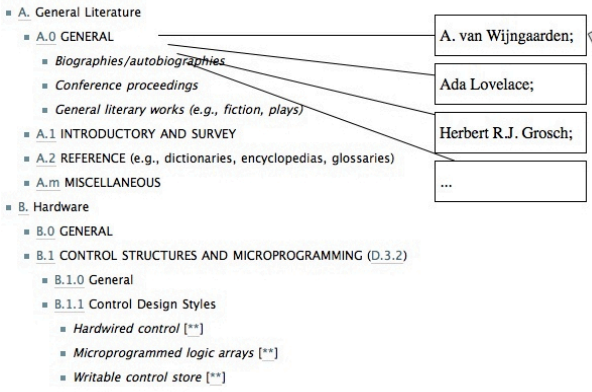


Figura 2.1 - Parte da árvore de classificação CSS98 [27]

Na construção do sistema considerou-se originalmente que o último nível, ou seja, as folhas ou *subject descriptors*, seria suficiente para lidar com alterações e avanços na área, sendo expectável que estivesse em constante mudança. No entanto, na prática, verificou-se a dificuldade de remover antigos *subject descriptors* sem perturbar referências para trabalhos classificados com esse tipo. Assim, assumiu-se que este nível é parte permanente da árvore de classificação, sendo identificados com asteriscos aqueles *subject descriptors* que se retiraram do uso activo [37]. No caso do exemplo CCS98 apresentado, apesar de não estar na figura, existem identificações com 1 ou 2 asteriscos, o que representam respectivamente classificações que se consideram sem uso após 1991 e após 1998, mas que se mantiveram para garantia de coerência e pesquisa.

De forma anexa a esta estrutura em árvore existem ainda 2 tipos de elementos, um conjunto de termos gerais (*General Terms*) e um conjunto adicional de *subject descriptors* referidos como *Implicit Subject Descriptors*. Estes termos gerais consistem num conjunto de 16 palavras (Ex: *Algorithms*, *Design*, *Documentation*, *Economics,* etc. [27]) que tipicamente se aplicam a várias áreas

dentro da CS e que são de certa forma ortogonais à árvore de classificação. Os *Implicit Subject Descriptors* (também referidos como *Proper Noun Subject Descriptors*) são efectivamente *subject descriptors* para nomes de produtos, linguagens e pessoas proeminentes da área da computação. Porém, estes não aparecem explicitamente na árvore de classificação, pois sendo uma lista demasiado extensa (actualmente 1719 elementos), a tornariam muito grande e complexa. Entendam-se como os "ramos" finais, associados aos vários nós da árvore de classificação, conforme a seguinte Figura 2.2 (esquerda). Mas que pela sua extensão (e repetições) estão numa tabela anexa [38], da qual se representa parte também na Figura 2.2 (direita). Naturalmente, o sistema de classificação assume que esta lista é dinâmica e refere expectáveis alterações frequentes sobre a mesma.



CCS Node	Noun
K.8.1	1-2-3/G
D.2.6	3DIS
D.3.2	8080 assembly language
G.3	A-STAT 79
K.2	A. van Wijngaarden
A.0	A. van Wijngaarden
K.2	ABC
D.3.2	ABC
D.3.2	ABCL/1
D.3.2	ABCL/R
D.3.2	ABEL
D.3.2	ABF
H.3.5	ABI/INFORM
K.2	ACE
D.3.2	ACLAN
K.7.2	ACM
D.3.4	ACORN

Figura 2.2 – Exemplo da relação da árvore CSS com os *Implicit Subject Descriptors* (esquerda) e parte da sua lista actual (direita).

A forma de aplicação do sistema de classificação, que está detalhada em instruções CCS98 – "*How to Use the Computing Classification System*" [39], apesar de ser relativamente intuitiva, contém alguns detalhes relevantes que se resumem de seguida. Só se aplica a classificação a nós abaixo do 1º nível, descendo o mais baixo possível na árvore, incluindo além dos *subject descriptors*, os *implicit subject descriptors* que se consideram relevantes. Só se deverá "parar" ao nível dos nós, caso nenhum dos *subject descriptors* desse nó se apliquem, ou caso todos se apliquem. A utilização de nós do tipo geral (*General*) só se deverá utilizar caso o trabalho a classificar cubra maioria dos conceitos dessa área (retratados nos nós "irmãos". Analogamente, só se deverá aplicar nós do tipo *Miscellaneous* quando o trabalho não seja possível de classificar em mais nenhum nó "irmão".

Seguindo as instruções, as classificações recolhidas deverão ser apresentadas respeitando o seguinte formato:

> ***Categories and subject descriptors**: Third-level node number [Second-level node title]: Third-level node title---first subject descriptor, second subject descriptor, etc.;*

Mostra-se um exemplo, referido pelo guia de utilização CSS [39]:

> "***Categories and subject descriptors**: D.2.7 [Software Engineering]: Distribution, Maintenance, and Enhancement---portability; H.2.3 [Database Management]: Languages---query languages; H.4.2 [Information Systems Applications]: Types of Systems---decision support; K.6.3 [Management of Computing and Information Systems]: Software Management---software selection*"

Por fim, deverão ainda ser considerados a aplicação dos 16 termos gerais (*General Terms*), bem como uma lista de palavras e frases-chave adicionais que não estejam retratadas no esquema de classificação, conforme os seguintes exemplos [39]:

> "***General Terms**: Design, Performance, Reliability*".
> "***Additional Key Words and Phrases**: databases, graph theory, inheritance conflicts, inheritance process, object-oriented database schemas, and recursive types*"

Tendo esta explicação detalhada do funcionamento do CCS, é importante ter em consideração alguns aspectos relativos à sua evolução, descritos no documento "*Computing Classification System 1998: Current Status and Future Maintenance*" [26]. Resumidamente, a quando da última revisão em 1998, são traçados alguns planos de manutenção e evolução do CCS, culminando em expectáveis alterações e revisões anuais. No entanto, quando se estabelecem recomendações para a revisão do CCS, é também referido que a estrutura base do CCS, bem como as metodologias e processos de suporte, têm sérios problemas que a comissão de revisão do CCS (em 98) não tinha capacidade para abordar. Assim, justifica-se o facto de terem sido apenas sugeridas em 1998 alterações aos níveis 2, 3 e 4 da árvore. Apesar de serem desejáveis alterações de fundo a todos os níveis (consolidações e renomeações), devido à evolução na área da CS e o facto de se ter detectado já algumas dificuldades de classificação da literatura da altura, tais não foram possíveis executar. Adicionalmente, sobre estas possíveis alterações, referem-se também problemas com a sua integração em software ACM (em utilização em 1998), relativos à compatibilização de versões, manutenção de histórico e integridade referencial do índice do sistema de classificação. Tais situações não eram possíveis de resolver com a curta alocação de *staff* disponível no momento. A nota final no documento é que se espera evolução continua sobre o CCS98, sendo necessário contínua intervenção de especialistas nas várias áreas da CS e contínua reestruturação dos níveis da árvore apoiada em técnicas automáticas (ex: análise estatística da literatura existente).

Apesar destas considerações, estando-se actualmente em 2009, não voltou a existir uma revisão do CCS (pelo menos nos 4 níveis base). Este facto, aliado aos

problemas acima mencionados para o actual CCS98, bem como a extensa evolução da área CS nos últimos 12 anos, levam à necessidade de estudos mais detalhados da sua actual adequabilidade e cobertura.

Porém, como forma de conclusão, tendo presente o funcionamento e aplicabilidade aqui descrita, o CCS não está necessariamente directamente relacionado com o trabalho da dissertação, i.e., a criação do modelo computável. Apesar de definir uma taxonomia relativamente abrangente para a área da CS, não é exactamente o *body of knowledge* sobre CS, detalhado o suficiente para permitir, por exemplo, uma definição de tópicos a abordar numa disciplina de um curso. A aplicabilidade do CCS no âmbito da dissertação, poderá eventualmente ser para o motivo que foi originalmente criado, ou seja, a classificação de material bibliográfico e publicações de consulta no âmbito das disciplinas do curso, a classificação de trabalhos produzidos em disciplinas mais avançadas (como o projecto), etc.

Apresentam-se nos subcapítulos seguintes breves descrições de trabalhos que aplicam ou utilizam o CCS e que poderão reforçar a conclusão acima. Ou seja, a sua eventual utilização na classificação de bibliografia e trabalhos produzidos no âmbito das disciplinas de um curso, mas não directamente como um *body of knowledge* sobre CS para um modelo computável de especificação de cursos.

2.1.1 Integração do ACM-CCS no DSpace

O departamento de sistemas de informação da universidade do Minho (DSI – UMINHO)[7] tem desenvolvido trabalho com a ferramenta *DSpace* [40] [41], um software *open source* para a construção de repositórios digitais, desenvolvido originalmente pelo *Massachusetts Institute of Technology* (MIT)[8] em parceria com os laboratórios Hewlett-Packard[9]. Essencialmente, o DSpace permite a recolha e índice de elementos digitais tais como artigos científicos, publicações técnicas, dissertações e teses, imagens, áudio e vídeo, utilizando o *standard Dublin Core* [42] para inclusão de meta-informação em cada elemento.

A sua utilização no DSI-UMINHO focou-se principalmente na criação de repositórios para a Universidade, tendo gerado alguns projectos de investigação e desenvolvimento como a tradução do *DSpace* para Português, alterações à versão original da componente de meta dados e a criação de um conjunto de *adejos* para extensão da sua versão base [41][43]. Este conjunto de *add-ons* servem vários propósitos, comunicação informal e (*add-on* de comentários), controlo de vocabulário utilizado na descrição do material (*add-on* de ontologias), sugestões de recursos relacionados (*add-on* de recomendação) e visualização de relações (*add-on* teia de comunicação).

[7] http://www.dsi.uminho.pt/

[8] http://web.mit.edu/

[9] http://www.hpl.hp.com/

O *add-on* de ontologias foi desenvolvido como forma de permitir "*ao administrador controlar as palavras-chave utilizadas pelos utilizadores para descrever itens durante o processo de submissão*" [43]. Paralelamente, e considerando a natureza classificativa do CCS, foi executado trabalho de transformação do CCS98 num formato reconhecido pelo add-on de ontologias, permitindo assim uma classificação no *DSpace* de trabalhos e projectos dos alunos do departamento de sistemas de informação[7] [44]. Este trabalho originou, até à data desta dissertação, uma versão XML do ACM-CCS98 (que é a actualmente distribuída no site ACM [27][43]) e uma versão RDF-Schema, ainda não disponível (por se encontrar em revisão [44]). Apresenta-se em seguida um excerto do ACM-CCS98 em versão XML e respectivo resultado da sua utilização real na Figura 2.3.

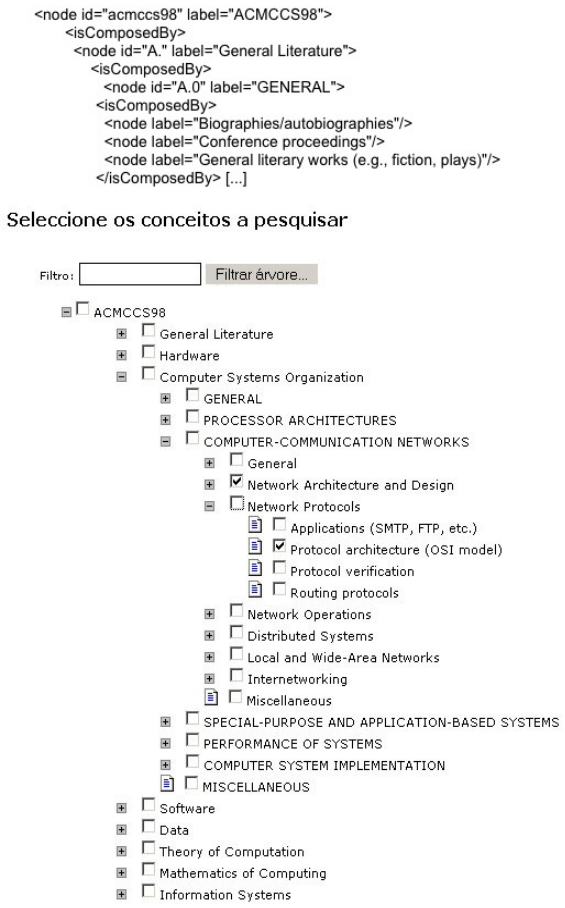


Figura 2.3 – Excerto da modelação XML do ACM-CSS98 (acima) e sua aplicação para classificação em *DSpace* (abaixo) [43]

2.1.2 Aplicação do ACM-CCS para classificação da investigação em Computer Science

Curiosamente, também na FCT-UNL se efectuou trabalho relacionado com a aplicação do CCS98. O projecto de investigação COPSRO – "*Computational Approach to Ontology Profiling of Scientific Research Organizations*", uma

colaboração do Centro de Inteligência Artificial (CENTRIA-UNL) com o *Department of Computer Science and Information Systems* do *Birkbeck College - University of London* (DCSIS-UL) e com o Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão do Instituto Superior de Engenharia do Instituto Politécnico do Porto (GECAD- ISEP), tem como principal objectivo o desenvolvimento de uma metodologia para estabelecer ontologias que permitam traçar o perfil e classificar organizações de pesquisa científica [45]. Pretende-se que a metodologia seja aplicada principalmente a organizações de pesquisa na área da CS, em especial nos departamentos de ciência da computação das universidades em Portugal e no Reino Unido, considerando como ontologia de referência o trabalho ACM-CCS (versão CCS98) [45] [46] [47].

O projecto envolve vários tipos de investigação, mas para esta dissertação, o relevante será o trabalho com o ACM-CCS, que essencialmente é utilizado como parte do problema exemplo a abordar. O CCS é utilizado como forma de classificação das actividades e temas de investigação das organizações, focando-se parte do projecto em procedimentos de simplificação e optimização da representação visual desta classificação (por exemplo aplicando técnicas *Cluster-Lift* [46] [47]). Refere-se em particular o trabalho "*Representing a Computer Science Research Organization on the ACM Computing Classification System*" [47] (no âmbito do projecto COPRSO), onde se efectuou uma experimentação prática dentro do Departamento de Informática da FCT-UNL (DI-FCT-UNL), relativa ao levantamento e classificação das actividades de investigação de 49 membros do departamento sobre o CCS98 (apenas ao 2º nível da árvore de classificação).

2.2 ACM/IEEE-CS/AIS *Computing Curricula*

Neste capítulo apresentam-se com algum detalhe os guias ACM/IEEE de cobertura temática e de proposta de estruturação curricular, CC2001 e CC2005. Como forma de introdução, descreve-se na secção 2.2.1 a evolução histórica dos vários *computing curricula* ACM, contextualizando o âmbito e objectivos do CC2001 e CC2005. Na secções seguintes, 2.2.2 e 2.2.3, abordam-se em detalhe e respectivamente, o CC2005 e o C2001. Pelas próprias naturezas dos relatórios, a secção relativa ao CC2005 foca-se na distinção das diversas áreas dentro da CS e respectivas competências a atingir, enquanto que a secção relativa ao CC2001, se foca nas estruturações curriculares propostas para o 1º ciclo CS.

2.2.1 Um pouco de história

Historicamente, a ACM e a IEEE-CS produziram vários *Curriculum Reports* para a área da computação, iniciando-se com o *Computing Curricula* 1991 (CC91) [48] que estabelecia um guia curricular para os cursos bacharelato de 4 anos em ciência da computação e engenharia de computadores, que foi sendo base para vários trabalhos específicos ao longo dos anos 90 (AssocDeg93, HS93, IS97, etc.) [31]. Com a evolução rápida da área da CS ao longo dos anos 90 e o surgimento

de muitas novas disciplinas e subáreas específicas (como a Engenharia de Software), detectou-se que esta diversidade, naturalmente transporta em muitos tipos de cursos disponíveis, gerava confusão e era um problema real. Como exemplo, apresenta-se a Figura 2.4, que representa como os potenciais alunos de cursos da área poderiam entender o seu foco, não compreendendo efectivamente quais as distinções entre *Computer Science* (CS), *Software Engineering* (SE) , *Computer Engineering* (CE), *Information Systems* (IS) e *Information Technology* (IT).

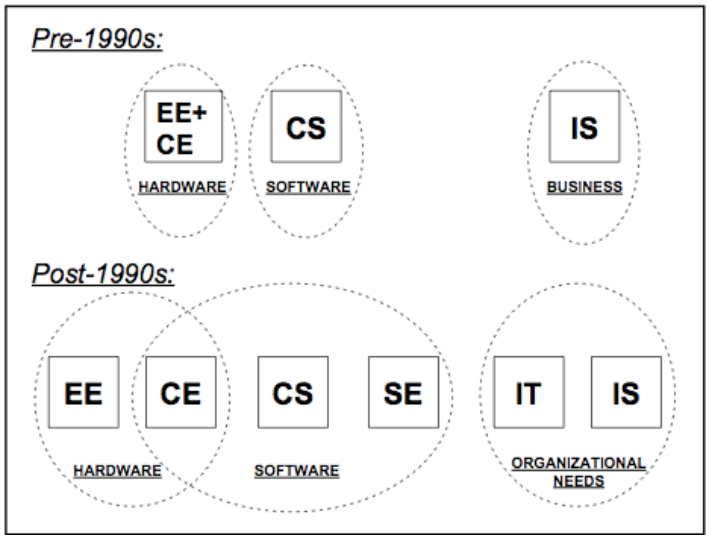


Figura 2.4 - Como os cursos da área (nos EUA) poderiam ser interpretados pelos potenciais alunos [31]

Quando no final dos anos 90, a ACM e o IEEE-CS voltaram a juntar-se para a produção de um novo *Curriculum Report* para substituição do CC91, perceberam que não podiam ignorar estes problemas. Concebeu-se então o plano original para o desenvolvimento do CC2001, que seria um único relatório com recomendações de estruturas curriculares para cursos das várias subáreas da computação.

No entanto, já durante os trabalhos de preparação do CC2001, verificou-se que devido à rápida expansão da área, um único relatório não seria suficiente. Assim, definiu-se que o CC2001 deveria suportar os currículos da principais subáreas, incluindo *Computer Engineering*, *Computer Science*, *Information Systems* e *Software Engineering*, mas também estar estruturalmente preparado para a inclusão de novas subáreas que eventualmente surgiriam. Esta necessidade de inclusão de novas subáreas e a necessidade de resolver o problema da confusão entre o foco das mesmas (e seus respectivos cursos), levou à existência de um relatório *Overview*, contextualizador dos vários relatórios das distintas subáreas [29] [31].

Porém, o grupo de trabalho ACM / IEEE-CS que preparava o CC2001 reconheceu que, sendo maioritariamente constituído por elementos da subárea

Computer Science, faria sentido desenvolver apenas o relatório específico para esta mesma. Assim, produziu-se aquilo que é hoje conhecido como *Computing Curricula* 2001 (ou CC2001) [29] e que, além do relatório com recomendações detalhadas para os currículos 1º ciclo da área CS, engloba também as visões do grupo de trabalho relativo à definição da estrutura para a produção de novos relatórios *Computing Curriculum*, conforme se apresenta na Figura 2.5.

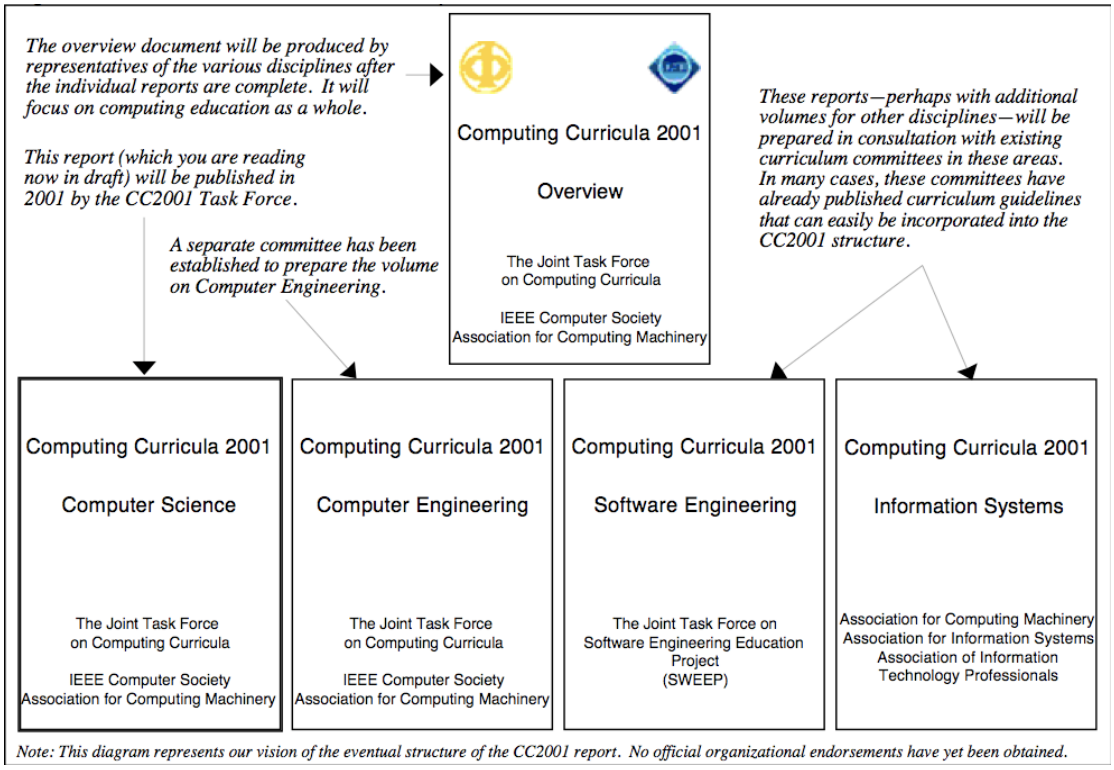


Figura 2.5 - Estrutura proposta no CC2001 para os *Computing Curricula*

Após a publicação do CC2001 foram sendo produzidos os outros relatórios específicos das várias subáreas, *Information Systems* em 2002 (IS2002) [49], *Software Engineering* em 2004 (SE2004) [50] e *Computer Engineering* em 2004 (CE2004) [51].

Finalmente em 2005, e já tendo algumas "pistas" sobre a evolução da área da computação em distintas subáreas, foi produzido o *Computing Curricula* 2005 – *The Overview Report* (CC2005) [31], que consubstancia o relatório agregador e contextualizador acima referido como necessário.

A previsão de evolução rápida da área e criação de novas subáreas, originalmente efectuada no CC2001, foi de certa forma já verificada no CC2005, onde se relata trabalho efectuado para a produção do relatório *Information Technology*, que tendo sido originalmente previsto para 2006, foi efectivamente apenas em 2008, ficando assim *Information Technology* 2008 (IT2008) [52]. Note-se que, pelo o facto do CC2001 incluir o currículo CS, poderia ser chamado CS2001, algo que deverá acontecer a quando do lançamento de novos currículos específicos CS.

Estando já em 2010, verificou-se que efectivamente o previsto IT2006 apesar de várias versões *draft* entre 2005 e 2007, só foi efectivamente fechado em 2008, produzindo assim o Até à data foram também produzidos trabalhos adicionais: Na subárea IS, o MSIS2006 [53] que é um relatório guia para o cursos IS do 2º ciclo; Na subárea CS, uma revisão provisória para actualização do CC2001, especificamente focada nas secções *Body of Knowledge*, produzindo assim o relatório CS2008 em Dezembro de 2008 [54]; E na área SE, o GSwE2009, um guia curricular revisto para os 2º ciclos da área SE [55].

Por fim, é importante perceber que a dissertação está focada no CC2001, pois foi iniciada antes da existência pública do CS2008. Devido ao facto deste ser uma revisão essencialmente focada na actualização de alguns temas (*Body of Knowledge*, maioritariamente) e não uma revisão estrutural completa, permite-se assim basear o trabalho conceptual na versão CC2001 que mantém o mesmo tipo de estruturas base. Um aspecto igualmente relevante é o facto do CC2001 estar também disponível em formatos computáveis utilizáveis (HTML) e o CS2008 não.

Estando o histórico e as relações entre os relatórios CC2001 e CC2005 contextualizadas, passa-se então à apresentação resumida do seu conteúdo.

2.2.2 ACM/IEEE-CS/AIS Computing Curricula 2005 – *Overview report*

O CC2005, como *Overview report* que é, está destinado principalmente às faculdades e universidades, administradores e outros membros da comunidade académica, pretendendo apoiar a obtenção de repostas a determinadas questões: Quais são os diferentes tipos de cursos na área da computação? Quais as suas diferenças e similaridades? O que o seus nomes efectivamente significam? Que cursos deverá a minha universidade oferecer? etc.

No relatório CC2005 descrevem-se textualmente cada uma das 5 subáreas principais, apresentando as suas diferenças com recurso a uma caracterização gráfica do espaço de problemas da área da computação ("*problem space of computing*"), conforme a Figura 2.6. Considera-se para a representação um cruzamento conceptual, sem suporte quantitativo efectivo (ou seja, um esboço com base na intuição e conhecimento dos autores), entre os assuntos e temas **da área da computação** onde tipicamente os alunos irão **trabalhar após o curso** (e não temas abordados no curso), com a sua respectiva aplicabilidade mais teórica ou prática.

O eixo vertical, referente aos temas da computação, contém os valores (de baixo para cima) "*Computer Hardware and Architecture*", "*Systems Infrastructure*", "*Software Methods and Technologies*", "*Application Technologies*" e "*Organizational Issues & Information Systems*", representando que o foco das áreas oscila entre o funcionamento interno dos aparelhos e as trocas de dados entre eles, até temas focados em informação, pessoas e aspectos organizacionais

do trabalho. O exemplo referido no CC2005 especifica que alguém que prefira o desenho de circuitos ou que tenha curiosidade sobre o funcionamento interno dos computadores, deverá focar-se na parte inferior do eixo. Enquanto que alguém que se preocupe com aspectos do impacto da tecnologia nas organizações, ou a sua melhor forma de aplicação para resolução dos problemas das pessoas e do trabalho, se deverá focar na parte superior do eixo. Por outro lado, o eixo horizontal, referente à aplicabilidade mais teórica ou mais prática dos temas, varia (da esquerda para a direita) apenas entre os valores "*Theory, Principles, Innovation*" e "*Application, Deployment, Configuration*". O exemplo citado pelo CC2005 especifica que alguém que tenha preferência pelo trabalho num laboratório para a invenção de algo novo, ou numa universidade para o desenvolvimento de novos princípios, deverá escolher o seu trabalho numa área que tenha uma maior ocupação do lado esquerdo. Enquanto que alguém que pretenda apoiar pessoas na escolha das tecnologias mais correctas ou que queria integrar produtos *off-the-shelf* para resolver problemas organizacionais, quererá uma área com maior ênfase no lado direito.

Naturalmente, este cruzamento "intuitivo", considera que poderão haver casos específicos de carreiras que não serão representadas pelos temas deste esquema. Mais ainda, representa apenas tópicos da área da computação, o que para os casos das subáreas CE e IS não é totalmente abrangente, pois ambas envolvem vários tópicos que estão fora da área da computação (por exemplo, electrónica e física mais específica, bem como aspectos de economia e gestão).

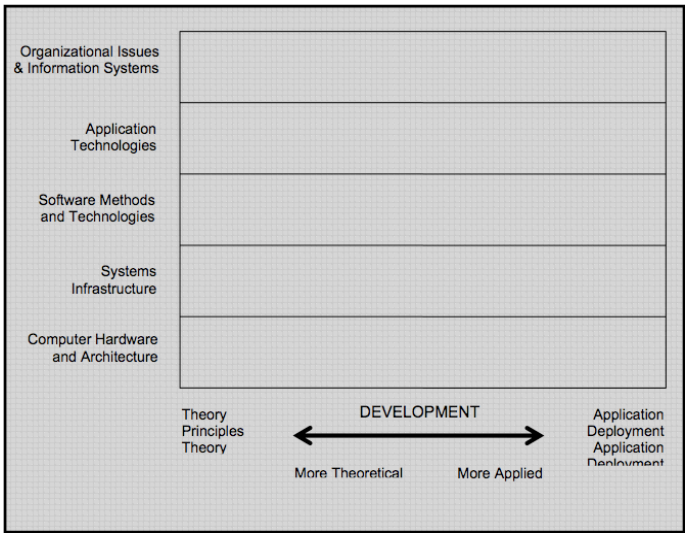


Figura 2.6 - Esboço "*Problem Space of Computing*" [31]

Identificam-se então as 5 subáreas principais (ou "*computing disciplines*"), descrevendo a sua caracterização conforme o entendimento dos autores CC2005 e apresentando-se os esquemas de cobertura no espaço de problemas da área da computação [31]. Um maior detalhe destes esquemas e suas descrições existe também no Anexo A.

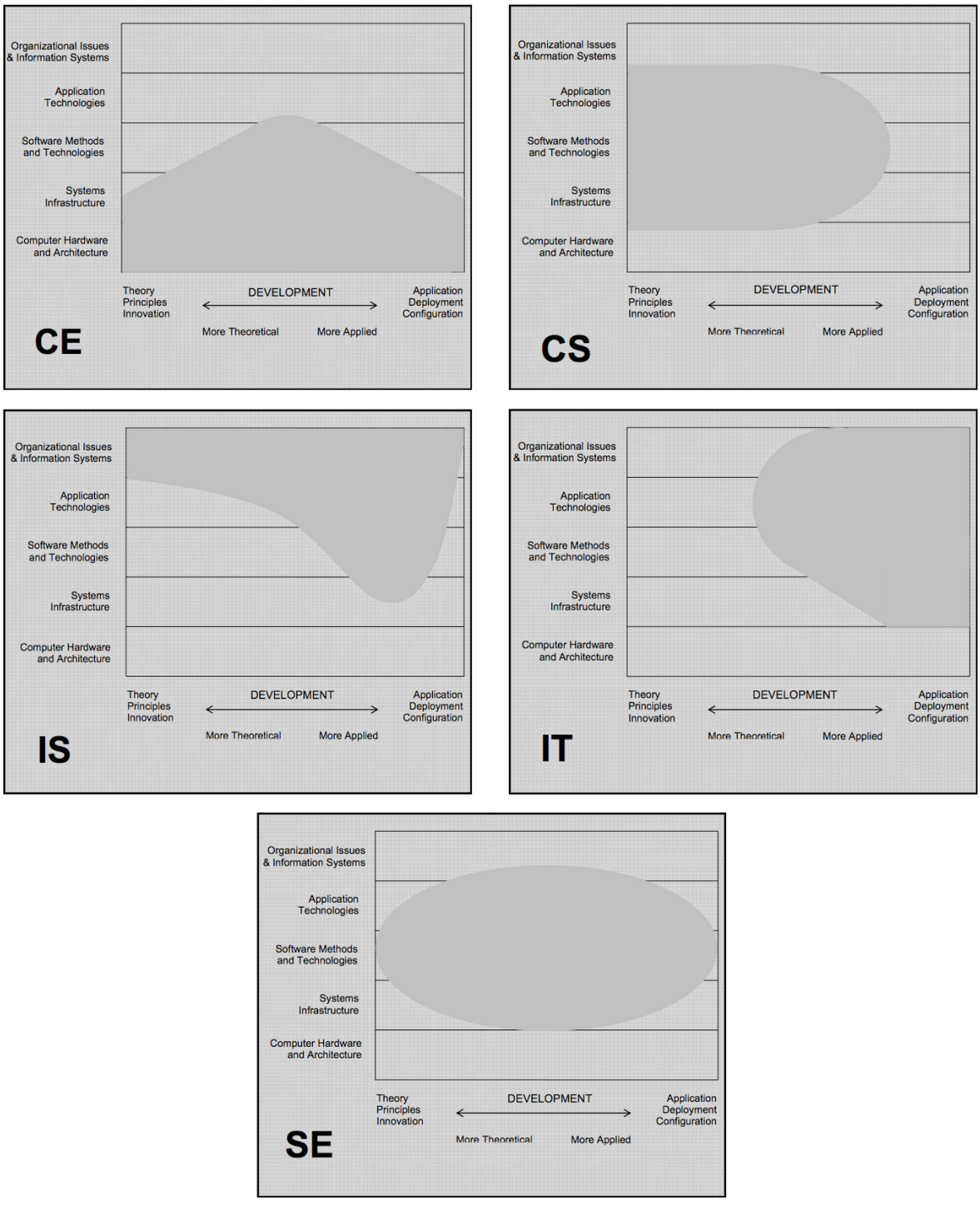


Figura 2.7 – Esquemas "*Problem Space of Computing*" das coberturas nas áreas CE, CS, IS, IT e SE [31]

A subárea de *Computer Engineering* (CE), foca-se na teoria e prática do desenvolvimento de hardware e software associado. O alto central no seu esquema, refere-se ao facto de que o interesse e aplicabilidade teórico-prática se foca quando se afasta do hardware e se aproxima do software. Um engenheiro desta área só necessitará de software o suficiente para suportar o seu desenvolvimento hardware.

A cobertura na subárea *Computer Science* (CS) é mais orientada ao meio (vertical) do gráfico pois não lida maioritariamente com hardware (para suporte ao software) nem aspectos organizacionais do uso de informação, focando-se geralmente em todos os outros aspectos. O gráfico estreita-se e pára em direcção à direita pois os trabalhadores CS não se focam no apoio às organizações na selecção dos produtos, na costumização às necessidades da organização ou no apoio à aprendizagem dos mesmos.

A representação da subárea *Information Systems* (IS) orienta-se mais ao topo do gráfico pois os trabalhadores IS focam-se mais nos aspectos das relações entre os sistemas de informação e as organizações, envolvendo-se também fortemente na implantação e configuração dos mesmos, com o subsequente apoio aos utilizadores. Note-se que, conforme já referido, a figura não inclui os tópicos IS dedicados à área de negócio e gestão.

No caso da subárea *Information Technology* (IT), o gráfico orienta-se mais ao lado direito pois esta foca-se mais nos aspectos de utilização, implantação e configuração das necessidades da organização e das suas pessoas, ao longo do espectro dos sistemas de informação da organização, desde a aplicação de tecnologia à infra-estrutura de sistemas. Foi identificado também que o desenvolvimento de cursos e programas educacionais nesta subárea, sendo relativamente recente, estão focados no fomentar de conceitos e *skills* existentes. Vários responsáveis pelos principais cursos IT consideraram em 2005 que a pesquisa na subárea irá eventualmente evoluir para novos temas e tópicos, podendo de futuro alterar a cobertura aqui apresentada.

Por fim, a subárea *Software Engineering* (SE) centra-se no meio do gráfico, o que representa um foco no desenvolvimento sistemático de software. O principal objectivo da SE é o desenvolvimento de modelos e técnicas para a produção de software de alto nível de qualidade, dentro do tempo e *budget*, o que implica uma cobertura quer dos aspectos teóricos, quer dos aspectos mais práticos.

Complementando esta caracterização alto-nível, que se acabou de apresentar, focada nas diferenças entre as subáreas, o CC2005 introduz também um sumário das características e comparação para **os cursos das várias subáreas**. Focam-se particularmente dois aspectos: o foco da cobertura temática dos cursos e as competências que os alunos deverão obter.

Consideraram-se 3 tabelas para representação das diferenças. As 2 primeiras comparam os tópicos e temas abordados nos cursos, respectivamente, tópicos específicos da área da computação e tópicos de outras áreas adicionais (não da computação) que também serão abordadas nos cursos. Veja-se o descrito na Figura 2.8 e Figura 2.9. A 3ª e última tabela, foca um conjunto de competências esperadas para os alunos que finalizam os cursos, atribuindo-lhes níveis de intensidade conforme expectativas de cada uma em cada curso.

Para as tabelas dos tópicos, considera-se um lista genérica, sumário da união dos vários assuntos abordados nas diferentes subáreas. Naturalmente, esta lista é uma visão generalizada, não correspondendo exactamente a todos os tópicos

abordados nos relatórios específicos para cada subárea, pois estes contém informação mais detalhada acerca dos temas abordados, utilizando por vezes nomenclaturas diferentes ou decompondo os tópicos em vários sub-tópicos. Considerou-se ainda que cada tópico deverá ter 2 classificações numéricas, um valor máximo e mínimo entre 0 e 5, correspondendo respectivamente ao ênfase mínimo e máximo que deverá ser dado nessa subárea a esse tópico, comparativamente com as outras subáreas.

Knowledge Area	CE		CS		IS		IT		SE	
	min	max	min	max	min	max	min	max	min	max
Programming Fundamentals	4	4	4	5	2	4	2	4	5	5
Integrative Programming	0	2	1	3	2	4	3	5	1	3
Algorithms and Complexity	2	4	4	5	1	2	1	2	3	4
Computer Architecture and Organization	5	5	2	4	1	2	1	2	2	4
Operating Systems Principles & Design	2	5	3	5	1	1	1	2	3	4
Operating Systems Configuration & Use	2	3	2	4	2	3	3	5	2	4
Net Centric Principles and Design	1	3	2	4	1	3	3	4	2	4
Net Centric Use and configuration	1	2	2	3	2	4	4	5	2	3
Platform technologies	0	1	0	2	1	3	2	4	0	3
Theory of Programming Languages	1	2	3	5	0	1	0	1	2	4
Human-Computer Interaction	2	5	2	4	2	5	4	5	3	5
Graphics and Visualization	1	3	1	5	1	1	0	1	1	3
Intelligent Systems (AI)	1	3	2	5	1	1	0	0	0	0
Information Management (DB) Theory	1	3	2	5	1	3	1	1	2	5
Information Management (DB) Practice	1	2	1	4	4	5	3	4	1	4
Scientific computing (Numerical mthds)	0	2	0	5	0	0	0	0	0	0
Legal / Professional / Ethics / Society	2	5	2	4	2	5	2	4	2	5
Information Systems Development	0	2	0	2	5	5	1	3	2	4
Analysis of Business Requirements	0	1	0	1	5	5	1	2	1	3
E-business	0	0	0	0	4	5	1	2	0	3
Analysis of Technical Requirements	2	5	2	4	2	4	3	5	3	5
Engineering Foundations for SW	1	2	1	2	1	1	0	0	2	5
Engineering Economics for SW	1	3	0	1	1	2	0	1	2	3
Software Modeling and Analysis	1	3	2	3	3	3	1	3	4	5
Software Design	2	4	3	5	1	3	1	2	5	5
Software Verification and Validation	1	3	1	2	1	2	1	2	4	5
Software Evolution (maintenance)	1	3	1	1	1	2	1	2	2	4
Software Process	1	1	1	2	1	2	1	1	2	5
Software Quality	1	2	1	2	1	2	1	2	2	4
Comp Systems Engineering	5	5	1	2	0	0	0	0	2	3
Digital logic	5	5	2	3	1	1	1	1	0	3
Embedded Systems	2	5	0	3	0	0	0	1	0	4
Distributed Systems	3	5	1	3	2	4	1	3	2	4
Security: issues and principles	2	3	1	4	2	3	1	3	1	3
Security: implementation and mgt	1	2	1	3	1	3	3	5	1	3
Systems administration	1	2	1	1	1	3	3	5	1	2
Management of Info Systems Org.	0	0	0	0	3	5	0	0	0	0
Systems integration	1	4	1	2	1	4	4	5	1	4
Digital media development	0	2	0	1	1	2	3	5	0	1
Technical support	0	1	0	1	1	3	5	5	0	1

Figura 2.8 - Comparação dos "pesos" entre tópicos especificamente da computação, para os cursos das várias subáreas [31]

Knowledge Area	CE		CS		IS		IT		SE	
	min	max	min	max	min	max	min	max	min	max
Organizational Theory	0	0	0	0	1	4	1	2	0	0
Decision Theory	0	0	0	0	3	3	0	1	0	0
Organizational Behavior	0	0	0	0	3	5	1	2	0	0
Organizational Change Management	0	0	0	0	2	2	1	2	0	0
General Systems Theory	0	0	0	0	2	2	1	2	0	0
Risk Management (Project, safety risk)	2	4	1	1	2	3	1	4	2	4
Project Management	2	4	1	2	3	5	2	3	4	5
Business Models	0	0	0	0	4	5	0	0	0	0
Functional Business Areas	0	0	0	0	4	5	0	0	0	0
Evaluation of Business Performance	0	0	0	0	4	5	0	0	0	0
Circuits and Systems	5	5	0	2	0	0	0	1	0	0
Electronics	5	5	0	0	0	0	0	1	0	0
Digital Signal Processing	3	5	0	2	0	0	0	0	0	2
VLSI design	2	5	0	1	0	0	0	0	0	1
HW testing and fault tolerance	3	5	0	0	0	0	0	2	0	0
Mathematical foundations	4	5	4	5	2	4	2	4	3	5
Interpersonal communication	3	4	1	4	3	5	3	4	3	4

Figura 2.9 – Comparação entre tópicos não-computacionais e respectivo "peso" para os cursos das várias subáreas [31]

Estes tópicos e respectivos "pesos" foram obtidos de forma consensual entre os autores do CC2005, tendo sido examinados os vários relatórios específicos de

cada subárea e aplicada a experiência e conhecimento prático de cada elemento da *task force*. É indicado explicitamente que se consideram este números como "*fuzzy*", mas que apesar disso, obteve-se validação da consistência dos mesmos por vários elementos representantes das várias subáreas.

Pode-se dizer que estas 2 tabelas representam aquilo que um aluno irá estudar durante o curso, o que significa que falta abordar as competências, ou seja, o que se espera de um aluno após o frequentar de um curso. Considerou-se então um conjunto de 60 competências, organizadas em 11 categorias, para as quais foram atribuídos valores de 0 a 5, com o seguinte significado: 0 corresponde a que exista nenhuma expectativa que a competência seja adquirida, enquanto 5 corresponde ao inverso, ou seja, a uma alta expectativa que a competência seja adquirida. Este valores são como os das tabelas anteriores, ou seja, números "*fuzzy*" obtidos de forma consensual pelos autores e validados por elementos das respectivas subáreas.

Esta tabela, representada na Figura 2.10, resume essencialmente que: ex-alunos CE devem saber desenhar e implementar sistemas que envolvem integração *hardware* – *software*; ex-alunos CS devem estar preparados para um abrangente conjunto de funções, envolvendo aspectos teóricos do desenvolvimento de *software*; ex-alunos IS devem ser capazes de analisar requisitos e processos de negócios, para especificarem e desenhares sistemas que cumpram os objectivos das organizações; ex-alunos IT deverão ser capazes de planear, implementar, configurar e manter a infra-estrutura computacional das organizações; por fim, ex-alunos SE devem ser capazes de executar e gerir actividades nas várias fases do ciclo de vida de sistemas e software de grande dimensão.

Naturalmente o CC2005 aborda mais pormenores ao longo do documento, como por exemplo, aspectos relacionados com a adopção dos cursos e subáreas nas várias instituições americanas, mas estes são menos relevantes para esta dissertação.

Tendo em consideração o anteriormente descrito, pode-se concluir dizendo que **o CC2005 foca-se efectivamente na identificação e caracterização sumária de alto nível das várias subáreas**, focando genericamente aspectos acerca dos **corpos de conhecimento e competências a atingir**. Portanto, na sequência, sabe-se que **os relatórios específicos de cada subárea conterão os detalhes e propostas relevantes para os corpos de conhecimento, as competências e respectivas definições curriculares**. Sabe-se também que o CC2001 foi o documento que deu origem ao CC2005, subsequentemente servindo de padrão aos outros relatórios de cada subárea. Apesar de estar assumido em 2005 como o relatório específico da subárea CS, este documento por ter sido elaborado anteriormente, contém também várias abordagens e propostas genéricas de estruturação curricular, além das propostas para o corpo de conhecimento, competências e disciplinas.

Area	Performance Capability	CE	CS	IS	IT	SE
Algorithms	Prove theoretical results	3	5	1	0	3
	Develop solutions to programming problems	3	5	1	1	3
	Develop proof-of-concept programs	3	5	3	1	3
	Determine if faster solutions possible	3	5	1	1	3
Application programs	Design a word processor program	3	4	1	0	4
	Use word processor features well	3	3	5	5	3
	Train and support word processor users	2	2	4	5	2
	Design a spreadsheet program (e.g., Excel)	3	4	1	0	4
	Use spreadsheet features well	2	2	5	5	3
	Train and support spreadsheet users	2	2	4	5	2
Computer programming	Do small-scale programming	5	5	3	3	5
	Do large-scale programming	3	4	2	2	5
	Do systems programming	4	4	1	1	4
	Develop new software systems	3	4	3	1	5
	Create safety-critical systems	4	3	0	0	5
	Manage safety-critical projects	3	2	0	0	5
Hardware and devices	Design embedded systems	5	1	0	0	1
	Implement embedded systems	5	2	1	1	3
	Design computer peripherals	5	1	0	0	1
	Design complex sensor systems	5	1	0	0	1
	Design a chip	5	1	0	0	1
	Program a chip	5	1	0	0	1
	Design a computer	5	1	0	0	1
Human-computer interface	Create a software user interface	3	4	4	5	4
	Produce graphics or game software	2	5	0	0	5
	Design a human-friendly device	4	2	0	1	3
Information systems	Define information system requirements	2	2	5	3	4
	Design information systems	2	3	5	3	3
	Implement information systems	3	3	4	3	5
	Train users to use information systems	1	1	4	5	1
	Maintain and modify information systems	3	3	5	4	3
Information management (Database)	Design a database mgt system (e.g., Oracle)	2	5	1	0	4
	Model and design a database	2	2	5	5	2
	Implement information retrieval software	1	5	3	3	4
	Select database products	1	3	5	5	3
	Configure database products	1	2	5	5	2
	Manage databases	1	2	5	5	2
	Train and support database users	2	2	5	5	2
IT resource planning	Develop corporate information plan	0	0	5	3	0
	Develop computer resource plan	2	2	5	5	2
	Schedule/budget resource upgrades	2	2	5	5	2
	Install/upgrade computers	4	3	3	5	3
	Install/upgrade computer software	3	3	3	5	3
Intelligent systems	Design auto-reasoning systems	2	4	0	0	2
	Implement intelligent systems	2	4	0	0	4
Networking and communications	Design network configuration	3	3	3	4	2
	Select network components	2	2	4	5	2
	Install computer network	2	1	3	5	2
	Manage computer networks	3	3	3	5	3
	Implement communication software	5	4	1	1	4
	Manage communication resources	1	0	3	5	0
	Implement mobile computing system	5	3	0	1	3
	Manage mobile computing resources	3	2	2	4	2
Systems Development Through Integration	Manage an organization's web presence	2	2	4	5	2
	Configure & integrate e-commerce software	2	3	4	5	4
	Develop multimedia solutions	2	3	4	5	3
	Configure & integrate e-learning systems	1	2	5	5	3
	Develop business solutions	1	2	5	3	2
	Evaluate new forms of search engine	2	4	4	4	4

Figura 2.10 – Comparação entre as competências expectáveis para os alunos que frequentaram cursos das diferentes subáreas [31]

Assim aborda-se de seguida o CC2001 em detalhe, pois será o mais adequado para apoiar a definição de um modelo base, eventualmente utilizável para a construção de um modelo computável de um curso da área CS.

2.2.3 ACM/IEEE-CS Computing Curricula 2001 (CC2001)

Estando já anteriormente contextualizado na secção 2.2, irá ser focado agora especificamente o conteúdo do CC2001 [29], que essencialmente, além de algumas recomendações sobre a estruturação possível dos volumes *Curriculum Computing* (que mais tarde deram origem ao CC2005), contém recomendações para o desenvolvimento das estruturas curriculares de cursos 1º ciclo da área CS.

Em resumo, os principais aspectos que o CC2001 introduz e que são relevantes para esta dissertação:

i. **O corpo de conhecimento CS**

Um dos principais aspectos do CC2001 é apresentar uma proposta para um corpo de conhecimento (em diante referido como a abreviatura BoK – *Body Of Knowledge*) para a áreas CS. Propõe uma estrutura hierárquica em árvore, considerando uma divisão do conhecimento em áreas, posteriormente divididas em unidades de conhecimento, estando estas subsequentemente divididas em tópicos de conhecimento. Foram estabelecidas 14 áreas, 132 unidades e 815 tópicos. Em diante, e de forma a evitar equívocos com outras definições apresentadas futuramente, os constituintes do *BoK* serão referidos pelos seus nomes em inglês *area*, *unit* e *topic*.

ii. **O corpo de conhecimento core para o 1º ciclo CS**

Dentro das 132 *units* o CC2001 estabelece um conjunto de 64 *units* consideradas como *core*, que foram definidas consensualmente como essenciais existirem em qualquer curso CS do 1º ciclo.

iii. **Os objectivos de aprendizagem**

Associado a cada *unit* foi igualmente introduzido o conceito de objectivos de aprendizagem (em diante referidos como *learning objectives* ou *LOs*), com o objectivo de promover a verificação da aprendizagem dos alunos (estão directamente relacionados com as competências a adquirir). Adicionalmente foram também definidos um conjunto de objectivos mais gerais, que todos os alunos dos cursos do 1º ciclo deverão atingir. Estes deverão ser entendidos como competências ou características mínimas que os alunos CS deverão ter após frequentarem os cursos.

iv. **Modelos curriculares**

De forma complementar ao corpo de conhecimento, o CC2001 apresenta diversas recomendações de estruturação curricular baseadas em experiências de utilização real nas universidades. Introduzem recomendações específicas para os níveis introdutórios e intermédios dos cursos, bem como para os tradicionais modelos universitários de ano / semestre.

v. **Descrição de disciplinas para os currículos**

Tendo por base o corpo de conhecimento, apresentam em detalhe um conjunto de 47 disciplinas (definindo a sua descrição, temas, objectivos, etc.) que fazem parte dos modelos curriculares anteriormente propostos. Identificam-se também (mas não se

detalham) cerca de 80 disciplinas adicionais, classificando-as como avançadas e para possível inclusão em cursos do 1º ciclo.

O corpo de conhecimento CS, descrito no capítulo 5 e detalhado no apêndice A do CC2001 [29] [30], é constituído na realidade pelos pontos *i*, *ii* e *iii* acima descritos, ou seja, as *areas*, *units* e *topics*, com identificação dos respectivos *learning objectives*, bem como quais *units* são parte do *core* CS. Por outro lado, a descrição das disciplinas CS está detalhada no apêndice B do CC2001 [29] [56] e refere-se ao ponto *v* acima descrito, contendo também parcialmente alguns aspectos do ponto *iv*. As recomendações para modelos e estruturação curricular, que perfazem o ponto *iv*, existe essencialmente nos capítulos 6, 7, 8 e 9. Por fim, é importante também referir o capítulo 11, onde se descrevem quais as principais características que deverão ser atingidas pelos alunos que saem graduados dos cursos CS, focando-se mais na perspectiva dos objectivos que os cursos pretendem atingir.

Os vários assuntos abordados no CC2001 são de extrema relevância para quem necessite de definir um curso da área CS (refere-se por exemplo, o ponto *iv*). No entanto, os elementos realmente interessantes para o foco desta dissertação são efectivamente o corpo de conhecimento e sua estrutura proposta, bem como o modelo sugerido para descrição das disciplinas (e de certa forma, também as descrições das 47 disciplinas recomendadas), existentes concretamente nos apêndice A e B. Um outro aspecto não menos importante do CC2001, é o facto de estar disponível num formato computável [28], o que facilitará uma eventual extracção e utilização desta informação.

2.2.3.1 CC2001 Appendix A – CS Body of Knowledge

Este elemento, que o longo do resto da dissertação irá ser referido como o *Body of Knowledge* ACM-A (devido ao facto de ser existir no apêndice A do documento ACM CC2001), além de ser simplesmente um corpo de conhecimento que representa um conjunto de conceitos, termos, etc., da área CS, deverá também ser encarado como um modelo para recolha desta mesma informação, que possa evoluir e ser "ligado" ou relacionado a outros aspectos das disciplinas e dos cursos.

Essencialmente o CC2001 estabelece uma hierarquia com 3 níveis, onde o 1º nível se refere a áreas de conhecimento (*areas*), o 2º nível a unidades de conhecimento dentro das áreas (*units*) e o 3º nível como tópicos de conhecimento dentro das unidades (*topics*). Cada unidade de conhecimento terá adicionalmente objectivos de aprendizagem (*learning objectives* ou LOs). Utilizando uma nomenclatura mais "livre", sem nenhuma definição específica, pode-se visualmente entender esta hierarquia como:

AREA → UNIT → (TOPIC | LearningObjectives)

Uma *área*, que é o topo da hierarquia do BoK CS ACM-A, entende-se por área de conhecimento dentro da ciência de computação. Como exemplo, a área de estudo de algoritmos e complexidade [59] [60], é representada no BoK ACM-A por um elemento com o nome "AL. *Algorithms and Complexity*", que englobará um conjunto de informação (*units* e *topics*) sobre essa mesma temática. Uma *area* é sempre constituída por *units*, que representam conceptualmente módulos temáticos individuais [29] dentro dessa área. Segundo o proposto no CC2001, estas são também a principal forma para indicação das temáticas a abordar dentro das disciplinas dos cursos. Como exemplo, dentro da área referida atrás, existem no BoK ACM-A várias unidades, "AL1. *Basic algorithmic analysis*", "AL2. *Algorithmic strategies*", etc.., que seguirão sempre a nomenclatura das 2 letras identificativas da sua *area* e um número. Veja-se Figura 2.11 um *overview* do BoK do apêndice A, onde se apresentam as 14 *areas* e respectivas 132 *units* [29].

De imediato se verifica na figura que há 2 pormenores interessantes, o facto de algumas *units* estarem sublinhadas e o facto de se referirem horas nas mesmas. O sublinhado indica que o CC2001 propõe que estas *units* sejam ***core units***, ou seja, aquelas em que há um consenso alargado que o seu material é essencial para todos os alunos que frequentam cursos do 1º ciclo na área CS, sendo portanto necessárias para todos os alunos de todos os cursos. Aquelas não sublinhadas não fazem parte do *core* e deve ser referidas como *elective units*. Isto não significa que são menos importantes, mas apenas que não houve um consenso nos autores CC2001 acerca da sua obrigatoriedade para todos os alunos e cursos. Note-se que as *core units* não são necessariamente referentes a disciplinas introdutórias, nem são suficientes para um currículo CS completo, sendo necessário incluir as outras *units* para se estabelecer um curso do 1º ciclo (de acordo com o CC2001, sua escolha variará naturalmente conforme o tipo de curso, área de foco, a instituição, etc.). Nas *core units* foram também identificadas **horas**, que corresponderão a uma **métrica de esforço** referente ao tempo mínimo necessário para abordar e apresentar os temas da *unit* numa aula de formato padrão. O CC2001 refere que se pretende que esta medida seja encarada mais para efeitos comparativos, não obrigado ou recomendado o formato aula padrão, salvaguardando que estas horas apenas reflectem o tempo mínimo para passagem de conhecimento ao aluno e que as mesma não incluem tempo consumido fora da aula, quer para o aluno na aprendizagem, quer para o professor, na produção de conteúdos. Refere-se com recomendação geral que o esforço fora das aulas é aproximadamente 3 vezes o tempo do consumido nas aulas.

DS. Discrete Structures (43 core hours)
- DS1. Functions, relations, and sets (6)
- DS2. Basic logic (10)
- DS3. Proof techniques (12)
- DS4. Basics of counting (5)
- DS5. Graphs and trees (4)
- DS6. Discrete probability (6)

PF. Programming Fundamentals (38 core hours)
- PF1. Fundamental programming constructs (9)
- PF2. Algorithms and problem-solving (6)
- PF3. Fundamental data structures (14)
- PF4. Recursion (5)
- PF5. Event-driven programming (4)

AL. Algorithms and Complexity (31 core hours)
- AL1. Basic algorithmic analysis (4)
- AL2. Algorithmic strategies (6)
- AL3. Fundamental computing algorithms (12)
- AL4. Distributed algorithms (3)
- AL5. Basic computability (6)
- AL6. The complexity classes P and NP
- AL7. Automata theory
- AL8. Advanced algorithmic analysis
- AL9. Cryptographic algorithms
- AL10. Geometric algorithms
- AL11. Parallel algorithms

AR. Architecture and Organization (36 core hours)
- AR1. Digital logic and digital systems (6)
- AR2. Machine level representation of data (3)
- AR3. Assembly level machine organization (9)
- AR4. Memory system organization and architecture (5)
- AR5. Interfacing and communication (3)
- AR6. Functional organization (7)
- AR7. Multiprocessing and alternative architectures (3)
- AR8. Performance enhancements
- AR9. Architecture for networks and distributed systems

OS. Operating Systems (18 core hours)
- OS1. Overview of operating systems (2)
- OS2. Operating system principles (2)
- OS3. Concurrency (6)
- OS4. Scheduling and dispatch (3)
- OS5. Memory management (5)
- OS6. Device management
- OS7. Security and protection
- OS8. File systems
- OS9. Real-time and embedded systems
- OS10. Fault tolerance
- OS11. System performance evaluation
- OS12. Scripting

NC. Net-Centric Computing (15 core hours)
- NC1. Introduction to net-centric computing (2)
- NC2. Communication and networking (7)
- NC3. Network security (3)
- NC4. The web as an example of client-server computing (3)
- NC5. Building web applications
- NC6. Network management
- NC7. Compression and decompression
- NC8. Multimedia data technologies
- NC9. Wireless and mobile computing

PL. Programming Languages (21 core hours)
- PL1. Overview of programming languages (2)
- PL2. Virtual machines (1)
- PL3. Introduction to language translation (2)
- PL4. Declarations and types (3)
- PL5. Abstraction mechanisms (3)
- PL6. Object-oriented programming (10)
- PL7. Functional programming
- PL8. Language translation systems
- PL9. Type systems
- PL10. Programming language semantics
- PL11. Programming language design

Note: The numbers in parentheses represent the minimum number of hours required to cover this material in a lecture format. It is always appropriate to include more.

HC. Human-Computer Interaction (8 core hours)
- HC1. Foundations of human-computer interaction (6)
- HC2. Building a simple graphical user interface (2)
- HC3. Human-centered software evaluation
- HC4. Human-centered software development
- HC5. Graphical user-interface design
- HC6. Graphical user-interface programming
- HC7. HCI aspects of multimedia systems
- HC8. HCI aspects of collaboration and communication

GV. Graphics and Visual Computing (3 core hours)
- GV1. Fundamental techniques in graphics (2)
- GV2. Graphic systems (1)
- GV3. Graphic communication
- GV4. Geometric modeling
- GV5. Basic rendering
- GV6. Advanced rendering
- GV7. Advanced techniques
- GV8. Computer animation
- GV9. Visualization
- GV10. Virtual reality
- GV11. Computer vision

IS. Intelligent Systems (10 core hours)
- IS1. Fundamental issues in intelligent systems (1)
- IS2. Search and constraint satisfaction (5)
- IS3. Knowledge representation and reasoning (4)
- IS4. Advanced search
- IS5. Advanced knowledge representation and reasoning
- IS6. Agents
- IS7. Natural language processing
- IS8. Machine learning and neural networks
- IS9. AI planning systems
- IS10. Robotics

IM. Information Management (10 core hours)
- IM1. Information models and systems (3)
- IM2. Database systems (3)
- IM3. Data modeling (4)
- IM4. Relational databases
- IM5. Database query languages
- IM6. Relational database design
- IM7. Transaction processing
- IM8. Distributed databases
- IM9. Physical database design
- IM10. Data mining
- IM11. Information storage and retrieval
- IM12. Hypertext and hypermedia
- IM13. Multimedia information and systems
- IM14. Digital libraries

SP. Social and Professional Issues (16 core hours)
- SP1. History of computing (1)
- SP2. Social context of computing (3)
- SP3. Methods and tools of analysis (2)
- SP4. Professional and ethical responsibilities (3)
- SP5. Risks and liabilities of computer-based systems (2)
- SP6. Intellectual property (3)
- SP7. Privacy and civil liberties (2)
- SP8. Computer crime
- SP9. Economic issues in computing
- SP10. Philosophical frameworks

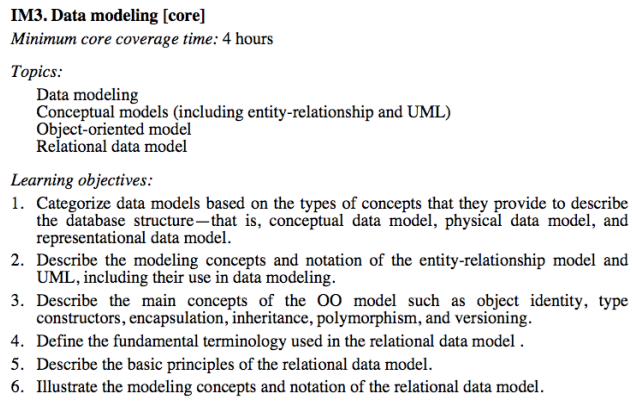
SE. Software Engineering (31 core hours)
- SE1. Software design (8)
- SE2. Using APIs (5)
- SE3. Software tools and environments (3)
- SE4. Software processes (2)
- SE5. Software requirements and specifications (4)
- SE6. Software validation (3)
- SE7. Software evolution (3)
- SE8. Software project management (3)
- SE9. Component-based computing
- SE10. Formal methods
- SE11. Software reliability
- SE12. Specialized systems development

CN. Computational Science (no core hours)
- CN1. Numerical analysis
- CN2. Operations research
- CN3. Modeling and simulation
- CN4. High-performance computing

Figura 2.11 – Body of Knowledge CS – *Areas* e *Units* (*core* a sublinhado)

A apresentação da informação no BoK está centrada à volta das *units*, estando as mesmas organizadas por secções correspondentes às *areas* respectivas. Cada

unit respeitará o formato representado na Figura 2.12, que utiliza como exemplo a *unit* "IM3. *Data Modeling*" da area "IM. *Information Management*".



IM3. Data modeling [core]

Minimum core coverage time: 4 hours

Topics:

Data modeling
Conceptual models (including entity-relationship and UML)
Object-oriented model
Relational data model

Learning objectives:

1. Categorize data models based on the types of concepts that they provide to describe the database structure—that is, conceptual data model, physical data model, and representational data model.
2. Describe the modeling concepts and notation of the entity-relationship model and UML, including their use in data modeling.
3. Describe the main concepts of the OO model such as object identity, type constructors, encapsulation, inheritance, polymorphism, and versioning.
4. Define the fundamental terminology used in the relational data model .
5. Describe the basic principles of the relational data model.
6. Illustrate the modeling concepts and notation of the relational data model.

Figura 2.12 – Bok ACM-A *unit* IM3. *Information Management*

Cada *unit* terá associado então, além da identificação se é *core* ou *elective* e as respectivas horas caso seja, um conjunto de *topics* e de *learning objectives*.

O conceito *topic*, que é o final da hierarquia BoK ACM, representa o "grão mais fino" do conhecimento CS e corresponde à divisão em tópicos temáticos da *unit* respectiva. Note-se que o CC2001 não considera nenhuma nomenclatura em particular para referenciar os *topic* dentro das *units*, sendo por isso viável assumir que o CC2001 considera que o nível mais baixo para referenciação externa será *unit*.

O conceito *learning objectives* (LOs), como o nome indica, representa objectivos de aprendizagem que, estando ligados a uma *unit* em particular, podem ser entendidos como capacidades, competências, *skills*[10], etc., que um aluno adquire, após frequentar ou concluir uma disciplina que aborde essa *unit* em particular. Além do exemplo da Figura 2.12 anterior, considere-se também *unit* "AL1", onde se indicam vários objectivos, "1. *Explain the use of big O, omega, and theta notation to describe the amount of work done by an algorithm*", "2. *Use big O, omega, and theta notation to give asymptotic upper, lower, and tight bounds on time and space complexity of algorithms*", etc. Note-se que inversamente ao conceito *topic*, os LOs estão identificados com números dentro de *unit*, sendo por isso possível referi-los externamente. Por fim, um aspecto bastante interessante a reter é o facto de rapidamente se perceber pelos seus textos, que estes englobam *skills* de diversos tipos, sejam de conhecimento (ou do "saber"), sejam práticas (ou do "fazer), ou mesmo *soft-skills*. Verifica-se então aqui uma ligação à orientação da definição dos cursos, conforme uma das preocupações de Bolonha, ou seja, a componente competência dos alunos.

[10] Conceito *Skill* - http://en.wikipedia.org/wiki/Skill

2.2.3.2 CC2001 *Appendix B – CS Course Descriptions*

O material apresentado neste apêndice B [56] tem como objectivo a definição de disciplinas de referência para os cursos CS. Descreve-se de forma detalhada, tendo por base a informação do apêndice A, um conjunto de 47 possíveis disciplinas para os cursos CS, referindo-se ainda, mas não detalhando, cerca de 80 disciplinas com teor avançado.

Apesar de não ser referido no CC2001 especificamente como *Body of Knowledge* (por motivos óbvios), o detalhe destas disciplinas estabelece uma base de informação muito útil relacionada com estruturação curricular, bem como, de certa forma, uma validação da utilização do material definido no BoK ACM-A. Assim, optou-se no âmbito desta dissertação, entender este material como um *Body of Knowledge* de estruturação curricular, usando em diante a referência **BoK ACM-B**. De forma similar ao apêndice A, e devido à sua estruturação da informação, este apêndice pode também ser interpretado como as bases de um modelo para a definição de disciplinas dos cursos da área CS, focado principalmente nos aspectos mais curriculares do posicionamento da disciplina, de cobertura temática, de cobertura de competências, etc. (por oposição aos aspectos mais operacionais com a avaliação, a bibliografia, etc.).

A apresentação desta informação sobre disciplinas utiliza os conceitos do BoK ACM-A, como as *areas*, *units*, *topics*, *core units* e respectivas horas, bem como alguns aspectos ainda não referidos neste documento, que fazem parte dos capítulos de estruturação curricular do CC2001. Em concreto, referem-se as propostas do CC2001 para a divisão de disciplinas em níveis e para as estratégias de organização / implementação da estrutura curricular.

O CC2001 adopta 3 níveis para as disciplinas, consoante o seu posicionamento no currículo: introdutórias, tipicamente disciplinas da temática base oferecidas no 1º ou 2º ano do curso; intermédias, tipicamente do 2º ou 3º ano e que estabelecem uma fundação base mais aprofundada para o estudo de determinado assunto; avançadas, tipicamente do último ano ou do 2º ciclo, focando-se em assuntos mais avançados que requerem tipicamente bases temáticas de anteriores disciplinas. Note-se que o CC2001, tendo sido elaborado em 2001 e maioritariamente por autores dos estados unidos (apesar de ter "preocupações" internacionais), tem habitualmente por base os cursos de 4 ou 5 anos (conforme o cenário anterior em Portugal), não forçando ou comprometendo-se no entanto exclusivamente com esse modelo. Por isto, e pelo facto do CC2001 se tentar focar na abrangência, os conceitos aqui apresentados pode ser ajustados e aplicados a cursos já compatíveis com Bolonha. No entanto, o CC2005, sendo apenas um *Overview Report* orientado a programas do 1º ciclo (ou *undergraduate degrees*), não expressa considerações acerca da duração dos cursos.

O CC2001 estabelece uma proposta para vários tipos de "estratégias de implementação" [29] e organização de disciplinas dentro da estrutura curricular. Este conceito, que é abordado em detalhe nos capítulos 6, 7, 8 e 9 do CC2001,

pode-se entender como uma lista de recomendações de distintas abordagens à estruturação curricular e em concreto, ao ensino e introdução à ciência da computação ou dos seus vários temas. Em cada estratégia (ou também referida como abordagem), propõe-se o agrupamento de diversos tipos de disciplinas, por diferentes sequências, focando-se assim com maior ou menor ênfase em determinados temas, mas sempre com o objecto de garantir os requisitos pedagógicos mínimos dos cursos. A Figura 2.13, apresenta a lista das diversas abordagens propostas, de acordo com os níveis das disciplinas dentro do curso.

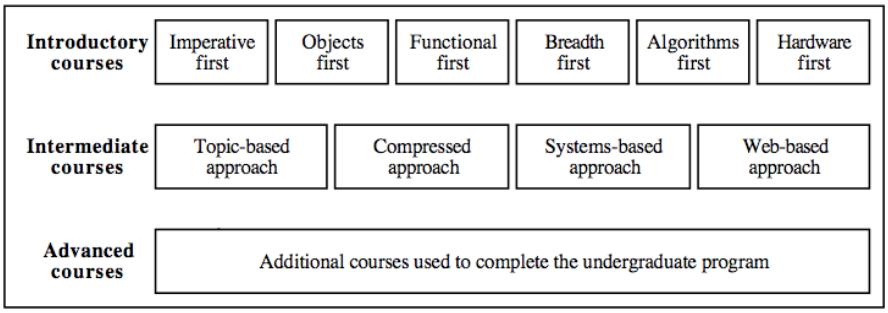


Figura 2.13 - Estratégias de implementação curriculares propostas no CC2001

Em concreto, cada estratégia de implementação estabelece uma diferente forma de apresentar ou introduzir os vários temas requisitos do curso, focando-se em diferentes abordagens ao ensino dos mesmos, estipulando para isso um conjunto bem identificado de disciplinas com uma orientação comum e respectiva proposta de sequência ou organização das mesmas nos semestres / anos do curso. Veja-se por exemplo, na Figura 2.14, a abordagem "*Imperative-first*", que é o modelo mais tradicional, onde se estipula um modelo em que o foco inicial do ensino da programação deverá ser feito pela abordagem das linguagens de programação imperativas no início.

Esta abordagem estipula duas possíveis versões, uma de 3 disciplinas e outra com apenas 2, correspondendo a 3 e 2 semestres respectivamente. Na versão 2 semestres (em baixo), a disciplina inicial foca-se na introdução da programação com uma abordagem imperativa, enquanto que a disciplina seguinte estende essa base, mas com um foco explícito no paradigma da programação por objectos. A versão de 3 semestres (em cima), expande o detalhe, o aprofundar e a possibilidade de adição de mais tópicos relativamente à versão anterior, permitindo que os alunos tenham melhor hipótese de dominar os conceitos fundamentais antes de avançar no curso.

CS101I. Programming Fundamentals
CS102I. The Object-Oriented Paradigm
CS103I. Data Structures and Algorithms

CS111I. Introduction to Programming
CS112I. Data Abstraction

Figura 2.14 - Abordagem curricular "*Imperative-First*" proposta no CC2001

Note-se que esta abordagem não impede que a 1ª disciplina use como exemplos, exercícios ou base de aprendizagem, uma linguagem de programação orientada a objectos. O que distingue esta abordagem, por exemplo da abordagem "*Object-First*" é o ênfase e ordenação dos tópicos iniciais em que a 1ª disciplina se deverá focar, ou seja, os aspectos imperativos da linguagem (como expressões, estruturas de controlo, procedimentos, funções, etc.). Os aspectos relativos ao desenho por objectos, deverão ser delegados para a(s) disciplina(s) seguinte(s).

Este conceito, que é também utilizando no apêndice B como uma forma de introduzir a classificação do "modelo pedagógico" da disciplina (ou seja, quais as estratégias a que a disciplina tipicamente pertence), é indicado apenas como recomendação, sendo inclusive sugerido a quando da definição curricular, que se experimentem as várias abordagem propostas, ou se pense na criação e inovação de novas aproximações.

Conforme se pôde ver no exemplo acima, o CC2001 apresenta também uma proposta para a numeração das disciplinas. Este aspecto é interessante pois introduz uma nomenclatura que ao mesmo tempo identifica características e classificações da disciplina, conforme se verifica na seguinte Figura 2.15.

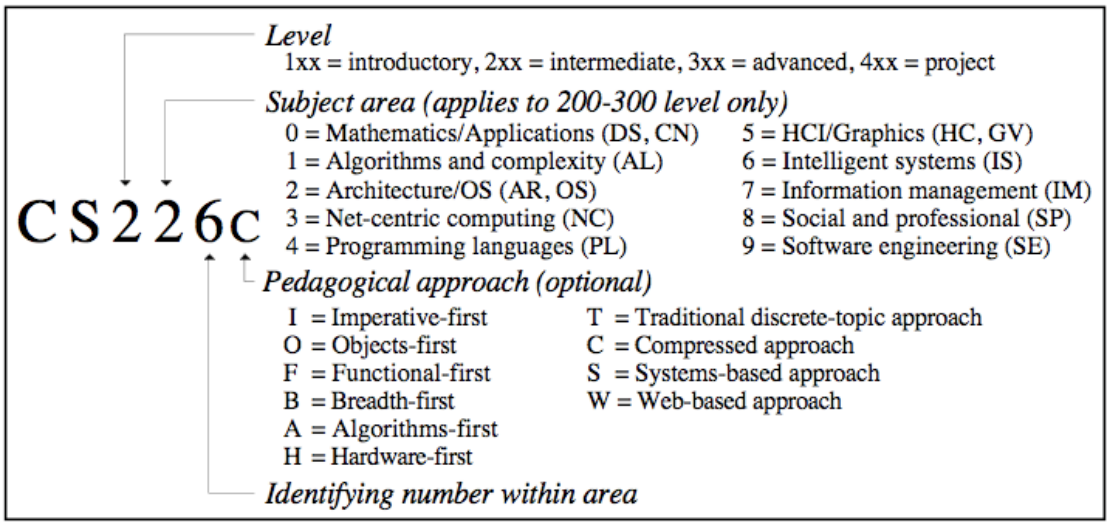


Figura 2.15 - Detalhe da nomenclatura CC2001 para numeração de *disciplinas*

Um pormenor que não é imediato na visualização desta nomenclatura é o facto do 2º número, indicado como *subject area* (e que também corresponde às *areas* do BoK ACM-A), só ser aplicado à cadeiras do nível intermédio e avançado ("2xx" e "3xx"). Isto é devido ao facto das diferentes abordagens pedagógicas do nível introdutório ("1xx"), poderem incluir distintas versões de 3 ou 2 disciplinas, sendo identificadas na nomenclatura respectivamente com um "0" e um "1" no 2º número ("10x" ou "11x"). Conforme se pode consultar no apêndice B [56], as abordagens *Imperative-first* e *Objects-first* ("I" e "O") contêm versões para 3 e 2 disciplinas, a abordagem *Breadth-first* ("B") contém apenas versões para 3 disciplinas (apesar sugerir 2 diferentes aproximações), e as abordagens

Functional-first, *Algorithms-first* e *Hardware-first* ("F", "A" e "H") contêm versões apenas para 2 disciplinas.

Por fim, o BoK ACM-B apresenta a informação das diferentes disciplinas seguindo o modelo representado na Figura 2.16, estabelecendo uma divisão por secções conforme os níveis e respectivas abordagens pedagógicas.

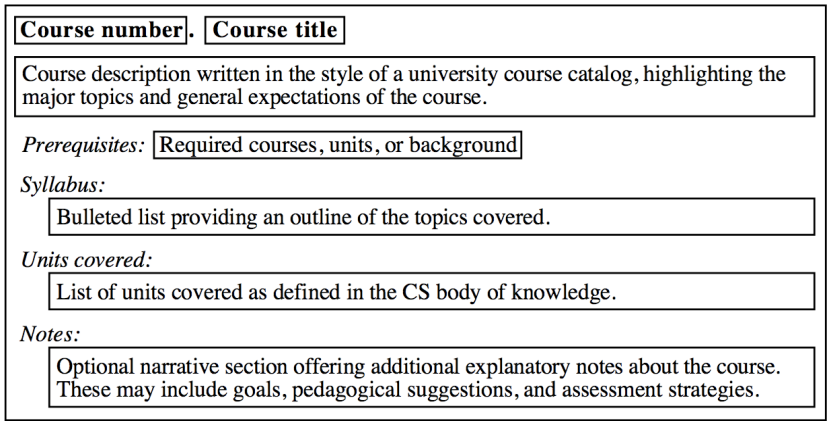
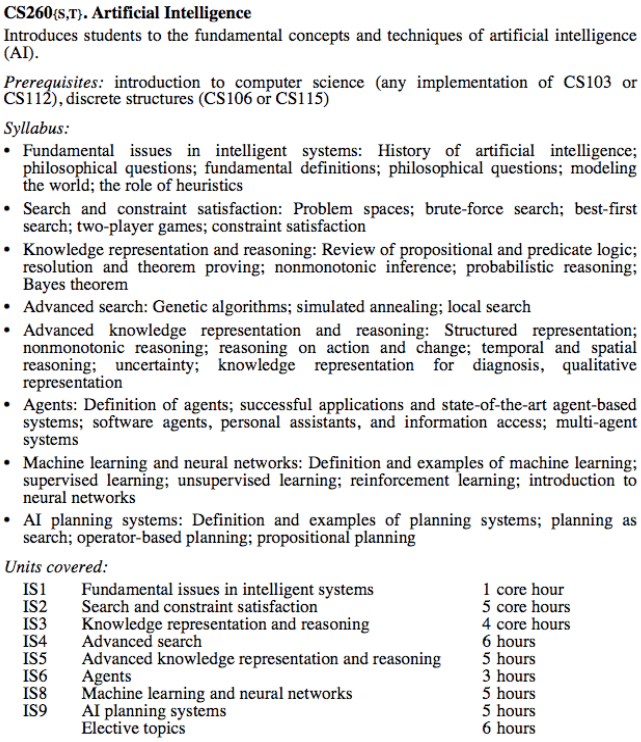


Figura 2.16 – Modelo CC2001 para a descrição de um *course*

O modelo é relativamente auto-explicativo, estando maioritariamente orientado à descrição textual. O *course number* segue a nomenclatura acima referida, enquanto que o *course title* e *course description* são puramente textuais. No *prerequisites*, apesar de também puramente textual, geralmente referem-se determinas áreas específicas do conhecimento, *courses* específicos através do número, ou eventualmente *units* do BoK ACM-A. Em *notes*, texto corrido geralmente opcional, estão indicadas notas genéricas para cobrir alguns detalhes específicos da disciplina. Em *units covered* estão explicitamente identificadas as *units* do BoK ACM-A que disciplina irá cobrir, **estando associado um valor indicativo das horas consumidas para abordar cada uma**. Por fim, um aspecto curioso é o facto do *syllabus* ser uma lista puramente textual que resume os tópicos a abordar na disciplina (e que permite alguma verbosidade), e não uma identificação específica de *topics* das respectivas *units* ACM-A. Note-se no entanto, que após uma verificação manual dos textos dos *syllabus* dos vários *courses* e dos respectivos *topics* das *units* referidas, percebe-se que os textos referem essencialmente o mesmo assunto (conceptualmente).

Veja-se um exemplo concreto na Figura 2.17, neste caso da disciplina "CS260{S,T}. *Artificial Intelligence*". Conforme se pode verificar pela nomenclatura deste número, as disciplinas intermédias poderão estar incluídas em várias abordagens, sendo estas referidas pelos seus identificadores (conforme Figura 2.15) separados por vírgulas e entre chavetas. Neste caso, esta disciplina poderá estar incluída nas abordagens "***S**ystems-based*" e "***T**raditional discrete-topic*".



CS260{S,T}. Artificial Intelligence

Introduces students to the fundamental concepts and techniques of artificial intelligence (AI).

Prerequisites: introduction to computer science (any implementation of CS103 or CS112), discrete structures (CS106 or CS115)

Syllabus:

- Fundamental issues in intelligent systems: History of artificial intelligence; philosophical questions; fundamental definitions; philosophical questions; modeling the world; the role of heuristics
- Search and constraint satisfaction: Problem spaces; brute-force search; best-first search; two-player games; constraint satisfaction
- Knowledge representation and reasoning: Review of propositional and predicate logic; resolution and theorem proving; nonmonotonic inference; probabilistic reasoning; Bayes theorem
- Advanced search: Genetic algorithms; simulated annealing; local search
- Advanced knowledge representation and reasoning: Structured representation; nonmonotonic reasoning; reasoning on action and change; temporal and spatial reasoning; uncertainty; knowledge representation for diagnosis, qualitative representation
- Agents: Definition of agents; successful applications and state-of-the-art agent-based systems; software agents, personal assistants, and information access; multi-agent systems
- Machine learning and neural networks: Definition and examples of machine learning; supervised learning; unsupervised learning; reinforcement learning; introduction to neural networks
- AI planning systems: Definition and examples of planning systems; planning as search; operator-based planning; propositional planning

Units covered:

IS1	Fundamental issues in intelligent systems	1 core hour
IS2	Search and constraint satisfaction	5 core hours
IS3	Knowledge representation and reasoning	4 core hours
IS4	Advanced search	6 hours
IS5	Advanced knowledge representation and reasoning	5 hours
IS6	Agents	3 hours
IS8	Machine learning and neural networks	5 hours
IS9	AI planning systems	5 hours
	Elective topics	6 hours

Figura 2.17 – Descrição da disciplina CS260{S,T} – *Artificial Intelligence*

Por fim, para dar uma visão da abrangência e detalhe desta lista de disciplinas do apêndice B do CC2001, apresenta-se na Figura 2.18 os nomes das 47 disciplinas introdutórias e intermédias, sendo completados na Figura 2.19 com os nomes das 80 disciplinas avançadas, ambas extraídas da versão HTML do CC2001 [28] [56].

2.2.3.3 CC2001 Capítulo 11 – "*Characteristics of CS Graduates*"

Introduz-se no capítulo 11, a visão ACM das principais características que se espera que alunos graduados dos cursos CS deverão possuir.

Estas características têm como propósito a definição de *standards* e limites, criando assim bases comparativas, para as competências que todos os graduados dos cursos CS são esperados atingir. De acordo com a visão ACM, as características aqui definidas cobrem, em larga medida, as mesmas bases do que objectivos curriculares, definidos de forma mais *macro* para os programas académicos. A diferença entre estes é primariamente de perspectiva. Olhar para os objectivos de um programa académico, em termos das características dos seus graduados, torna mais fácil definir *standards* e medidas que permitem, de certa foram, verificar se este objectivos estão a ser cumpridos.

B.1 Introductory tracks
In the course descriptions that follow, the introductory tracks ar

B.1.1 Imperative-first
The imperative-first approach offers two separate implementatic

CS101I. Programming Fundamentals
CS102I. The Object-Oriented Paradigm
CS103I. Data Structures and Algorithms
CS111I. Introduction to Programming
CS112I. Data Abstraction

B.1.2 Objects-first
Like the imperative-first approach, the objects-first strategy is a

CS101O. Introduction to Object-Oriented Programming
CS102O. Objects and Data Abstraction
CS103O. Algorithms and Data Structures
CS111O. Object-Oriented Programming
CS112O. Object-Oriented Design and Methodology

B.1.3 Functional-first
The functional-first approach exists only in the two-semester fo

CS111F. Introduction to Functional Programming
CS112F. Objects and Algorithms

B.1.4 Breadth-first
As outlined in Chapter 7, we propose two implementations of a I
sequence (CS101B-102B-103B) so that there is time for the adc

CS100B. Preview of Computer Science
CS101B. Introduction to Computer Science
CS102B. Algorithms and Programming Techniques
CS103B. Principles of Object-Oriented Design

B.1.5 Algorithms-first
The algorithms-first approach exists only in the two-semester f

CS111A. Introduction to Algorithms and Applications
CS112A. Programming Methodology

B.1.6 Hardware-first
The hardware-first approach exists only in the two-semester fo

CS111H. Introduction to the Computer
CS112H. Object-Oriented Programming Techniques

B.2 Other first-year courses
The courses in this section are arranged in numerical order.

CS105. Discrete Structures I
CS106. Discrete Structures II
CS115. Discrete Structures for Computer Science
CS120. Introduction to Computer Organization
CS130. Introduction to the World-Wide Web

B.3 Intermediate courses
Although the courses in this section are typically identified witl
one track, all appropriate suffixes are shown.

CS210[C,S,T,W]. Algorithm Design and Analysis
CS220[C,S,T]. Computer Architecture
CS221W. Architecture and Operating Systems
CS222W. Architectures for Networking and Communication
CS225[S,T]. Operating Systems
CS226[C,S]. Operating Systems and Networking
CS230[T,W]. Net-centric Computing
CS240S. Programming Language Translation
CS250W. Human-Computer Interaction
CS255[S,W]. Computer Graphics
CS260[S,T]. Artificial Intelligence
CS261W. AI and Information
CS262C. Information and Knowledge Management
CS270T. Databases
CS271S. Information Management
CS280T. Social and Professional Issues
CS290T. Software Development
CS291S. Software Development and Systems Programming
CS292[C,W]. Software Development and Professional Practice

Figura 2.18 – Lista de disciplinas introdutórias e intermédias referidas no apêndice B do CC2001

O CC2001 refere que as características concretas que se espera que alunos graduados tenham, estão directamente relacionados com objectivos de aprendizagem das várias *core units* (referidos na secção 2.2.3.1, relativa ao BoK ACM-A,). Ou seja, efectivamente aquilo que um estudante deverá saber no final da aprendizagem de determinada *unit*. No entanto, nesta secção o objectivo é identificar-se não as características concretas, mas sim as **expectativas** que se atribuem aqueles que finalizam o curso CS com sucesso, algo que permite reflectir o sucesso estudantil a um nível mais global. Mais em concreto, o CC2001 refere que é esperado que os alunos de CS desenvolvam um conjunto largo de capacidades e competências, algumas específicas do curso CS, enquanto outras, mais genéricas, são expectáveis a qualquer graduado de um curso técnico. O CC2001 propõe uma divisão destas em 3 categorias gerais: *i*) Capacidades cognitivas, relativas a tarefas intelectuais específicas à área CS; *ii*) Competências práticas relacionadas com CS; *iii*) outras competências, que apesar de

desenvolvidas no contexto CS, são de natureza mais geral e portanto aplicáveis também noutros contextos; conforme se apresenta na Figura 2.20.

B.4 Advanced courses

The CC2001 Task Force has decided not to include in the printed report full descriptions of the advanced cou accessible from the CC2001 web page. A list of the advanced courses we propose appears in Figure B-4.

Figure B-4. Advanced courses by area

Discrete Structures (DS)
- CS301. Combinatorics
- CS302. Probability and Statistics
- CS303. Coding and Information Theory

Computational Science (CN)
- CS304. Computational Science
- CS305. Numerical Analysis
- CS306. Operations Research
- CS307. Simulation and Modeling
- CS308. Scientific Computing
- CS309. Computational Biology

Algorithms and Complexity (AL)
- CS310. Advanced Algorithmic Analysis
- CS311. Automata and Language Theory
- CS312. Cryptography
- CS313. Geometric Algorithms
- CS314. Parallel Algorithms

Architecture and Organization (AR)
- CS320. Advanced Computer Architecture
- CS321. Parallel Architectures
- CS322. System on a Chip
- CS323. VLSI Development
- CS324. Device Development

Operating Systems (OS)
- CS325. Advanced Operating Systems
- CS326. Concurrent and Distributed Systems
- CS327. Dependable Computing
- CS328. Fault Tolerance
- CS329. Real-Time Systems

Net-Centric Computing (NC)
- CS330. Advanced Computer Networks
- CS331. Distributed Systems
- CS332. Wireless and Mobile Computing
- CS333. Cluster Computing
- CS334. Data Compression
- CS335. Network Management
- CS336. Network Security
- CS337. Enterprise Networking
- CS338. Programming for the World-Wide Web

Programming Languages (PL)
- CS340. Compiler Construction
- CS341. Programming Language Design
- CS342. Programming Language Semantics
- CS343. Programming Paradigms
- CS344. Functional Programming
- CS345. Logic Programming
- CS346. Scripting Languages

Human-Computer Interaction (HC)
- CS350. Human-Centered Design and Evaluation
- CS351. Graphical User Interfaces
- CS352. Multimedia Systems Development
- CS353. Interactive Systems Development
- CS354. Computer-Supported Cooperative Work

Graphics and Visual Computing (GV)
- CS355. Advanced Computer Graphics
- CS356. Computer Animation
- CS357. Visualization
- CS358. Virtual Reality
- CS359. Genetic Algorithms

Intelligent Systems (IS)
- CS360. Intelligent Systems
- CS361. Automated Reasoning
- CS362. Knowledge-Based Systems
- CS363. Machine Learning
- CS364. Planning Systems
- CS365. Natural Language Processing
- CS366. Agents
- CS367. Robotics
- CS368. Symbolic Computation
- CS369. Genetic Algorithms

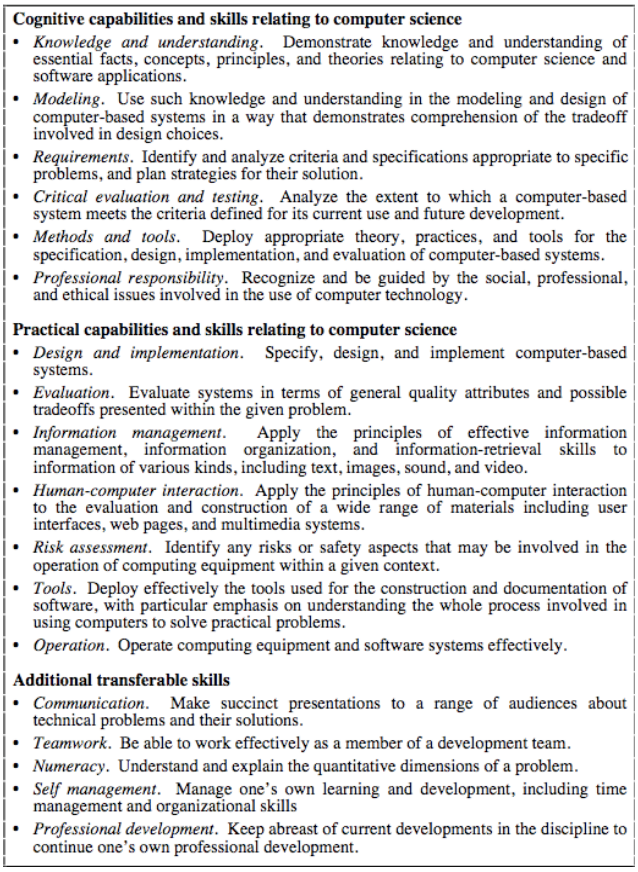
Information Management (IM)
- CS370. Advanced Database Systems
- CS371. Database Design
- CS372. Transaction Processing
- CS373. Distributed and Object Databases
- CS374. Data Mining
- CS375. Data Warehousing
- CS376. Multimedia Information Systems
- CS377. Digital Libraries

Social and Professional Issues (SP)
- CS380. Professional Practice
- CS381. Social Context of Computing
- CS382. Computers and Ethics
- CS383. Computing Economics
- CS384. Computer Law
- CS385. Intellectual Property
- CS386. Privacy and Civil Liberties

Software Engineering (SE)
- CS390. Advanced Software Development
- CS391. Software Engineering
- CS392. Software Design
- CS393. Software Engineering and Formal Specification
- CS394. Empirical Software Engineering
- CS395. Software Process Improvement
- CS396. Component-Based Computing
- CS397. Programming Environments
- CS398. Safety-Critical Systems

Figura 2.19 – Lista de disciplinas avançadas referidas no apêndice B do CC2001

Adicionalmente à proposta de um conjunto base de características expectáveis, o CC2001 apresenta ainda como recomendação um conjunto de medidas para determinar diferentes níveis de sucesso. Na Figura 2.21 apresentam-se o conjunto de objectivos mínimos (ou limite mínimo) que se espera que todos os alunos graduados deverão cumprir, bem como um *standard* que se espera de um aluno "médio".



Cognitive capabilities and skills relating to computer science

- *Knowledge and understanding*. Demonstrate knowledge and understanding of essential facts, concepts, principles, and theories relating to computer science and software applications.
- *Modeling*. Use such knowledge and understanding in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.
- *Requirements*. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.
- *Critical evaluation and testing*. Analyze the extent to which a computer-based system meets the criteria defined for its current use and future development.
- *Methods and tools*. Deploy appropriate theory, practices, and tools for the specification, design, implementation, and evaluation of computer-based systems.
- *Professional responsibility*. Recognize and be guided by the social, professional, and ethical issues involved in the use of computer technology.

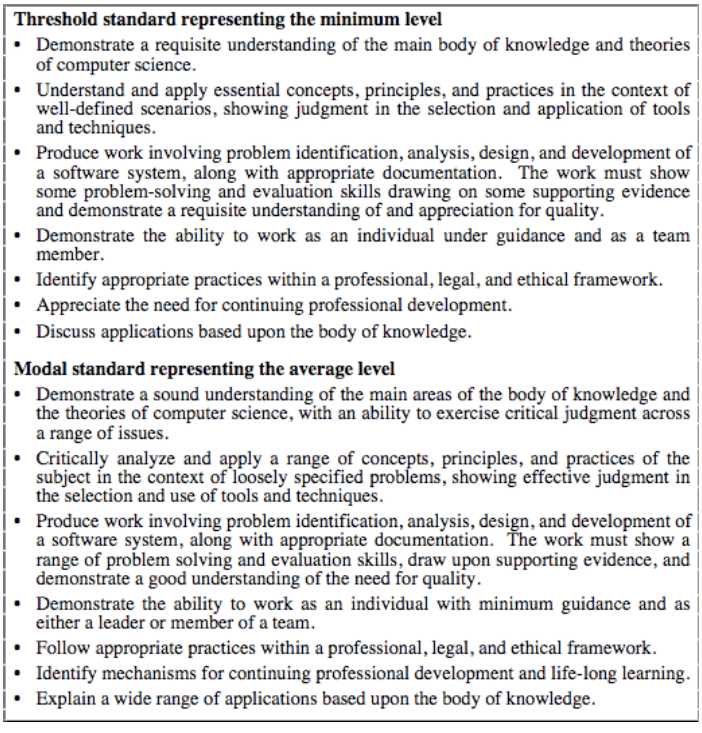
Practical capabilities and skills relating to computer science

- *Design and implementation*. Specify, design, and implement computer-based systems.
- *Evaluation*. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.
- *Information management*. Apply the principles of effective information management, information organization, and information-retrieval skills to information of various kinds, including text, images, sound, and video.
- *Human-computer interaction*. Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, and multimedia systems.
- *Risk assessment*. Identify any risks or safety aspects that may be involved in the operation of computing equipment within a given context.
- *Tools*. Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems.
- *Operation*. Operate computing equipment and software systems effectively.

Additional transferable skills

- *Communication*. Make succinct presentations to a range of audiences about technical problems and their solutions.
- *Teamwork*. Be able to work effectively as a member of a development team.
- *Numeracy*. Understand and explain the quantitative dimensions of a problem.
- *Self management*. Manage one's own learning and development, including time management and organizational skills
- *Professional development*. Keep abreast of current developments in the discipline to continue one's own professional development.

Figura 2.20 - Capacidades e competências expectáveis para alunos graduados com sucesso em cursos CS



Threshold standard representing the minimum level

- Demonstrate a requisite understanding of the main body of knowledge and theories of computer science.
- Understand and apply essential concepts, principles, and practices in the context of well-defined scenarios, showing judgment in the selection and application of tools and techniques.
- Produce work involving problem identification, analysis, design, and development of a software system, along with appropriate documentation. The work must show some problem-solving and evaluation skills drawing on some supporting evidence and demonstrate a requisite understanding of and appreciation for quality.
- Demonstrate the ability to work as an individual under guidance and as a team member.
- Identify appropriate practices within a professional, legal, and ethical framework.
- Appreciate the need for continuing professional development.
- Discuss applications based upon the body of knowledge.

Modal standard representing the average level

- Demonstrate a sound understanding of the main areas of the body of knowledge and the theories of computer science, with an ability to exercise critical judgment across a range of issues.
- Critically analyze and apply a range of concepts, principles, and practices of the subject in the context of loosely specified problems, showing effective judgment in the selection and use of tools and techniques.
- Produce work involving problem identification, analysis, design, and development of a software system, along with appropriate documentation. The work must show a range of problem solving and evaluation skills, draw upon supporting evidence, and demonstrate a good understanding of the need for quality.
- Demonstrate the ability to work as an individual with minimum guidance and as either a leader or member of a team.
- Follow appropriate practices within a professional, legal, and ethical framework.
- Identify mechanisms for continuing professional development and life-long learning.
- Explain a wide range of applications based upon the body of knowledge.

Figura 2.21 – *Standards*, limites ou objectivos expectáveis dos alunos graduados dos cursos CS

Apesar desta proposta, o CC2001 refere que estes medidas de *benchmarking* são definidas apenas para os níveis mínimos e médios, sendo no entanto importante que os cursos proporcionem também oportunidades para que alunos mais brilhantes atinjam o seu total potencial, referindo-se ao facto que os *curriculum* não deverão limitar aqueles que poderão ser os "líderes" no desenvolvimento da disciplina CS no futuro.

2.3 Aplicações do ACM/IEEE-CS Computing Curricula 2001

Com base nesta apresentação dos trabalhos IEEE/ACM, percebeu-se que o CC2001 é de especial interesse para esta dissertação. Particularmente no aspecto da proposta de modelos base para um corpo de conhecimento e para especificação da descrição curricular de disciplinas, mas também pelo facto de ter um largo conjunto de informação a suportar e validar os mesmos modelos.

Este interesse no CC2001 é também naturalmente partilhado por outros trabalhos, que abordam problemas similares ou relacionados com a criação de uma modelação para definição curricular. Após alguma investigação, encontraram-se vários trabalhos, alguns que apesar de vagamente relacionados com o tema desta dissertação não foram considerados relevantes para serem aqui detalhados em pormenor, enquanto outros, foram considerados relevantes o suficiente com o tema desta dissertação, para serem apresentados e sumariados nas subsecções seguintes.

Como trabalhos considerados menos relevantes para a sua sumarização, referem-se por exemplo, o trabalho "*Curriculum Development for Digital Libraries*" [57], que explora a necessidade da definição curricular na área particular das bibliotecas digitais (entenda-se do seu ensino) utilizando o CC2001 como base para o seu desenvolvimento, e o trabalho "*Knowledge management in an e-learning system*" [58], que propõe uma arquitectura para melhor explorar o conceito *e-learning*, considerando o CC2001 como uma ontologia para expressar meta-informação específica do domínio dos artefactos *e-learning*.

2.3.1 "*Deriving Ontology-based Metadata for E-learning from the ACM Computing Curricula*"

O 1º trabalho italiano, "*Deriving Ontology-based Metadata for E-learning from the ACM Computing Curricula*" [61], da universidade de Trento, foca-se principalmente na construção de uma ontologia para eventual definição de metadados [63] classificativos sobre material de suporte à aprendizagem da área CS. Uma parte do trabalho, concretamente a que interessa a esta dissertação, propõe que a construção dessa ontologia poderá ter como base o trabalho feito pelo IEEE/ACM, em particular o CC2001. Apesar de se indicar que o CC2001 não teve como objectivos a construção de uma ontologia da área CS, mas sim a construção de uma base para suporte à definição curricular, este trabalho propõe

que o corpo de conhecimento e descrição das disciplinas (apêndice A e B), poderá constituir a base para uma ontologia da área CS. Efectuou-se assim uma extracção da informação dos apêndices do CC2001 directamente do PDF [29], com recurso a um *parser* específico que produz um ficheiro XML [64] intermédio, e que é posteriormente limpo de incorrecções e ajustado. Como resultado final, e apoiado por um DTD [65] construído para validação da sintaxe, é produzido um ficheiro XML final que reproduz a estrutura do CC2001. Para apoiar os outros objectivos do trabalho, este XML foi também convertido num formato DAML+OIL [66] (um precursor da linguagem OWL [67]) para representação da ontologia, que representa então um mapeamento directo do modelo implícito no CC2001.

Este trabalho é interessante pois suporta a possibilidade da extracção da informação do CC2001 e sua utilização como modelo base. Apesar de estes ficheiros estarem indicados como disponíveis neste trabalho [61], no momento de elaboração desta dissertação já não os foi possível obter, existindo na altura como alternativa a versão HTML do CC2001.

2.3.2 "*Hyperkrep academic communities: an ontology-based approach to content portability*"

O 2º trabalho italiano, "*Hyperkrep academic communities: an ontology-based approach to content portability*" [62], este da universidade de Milão, utiliza alguns dos esforços do 1º trabalho e enquadra-se no âmbito de um sistema para distribuição de objectos de aprendizagem, suportado por uma representação partilhada do conhecimento semântico sobre a área específica. Especificamente para o âmbito desta dissertação, este trabalho contém outro estudo sobre uma possível abordagem à construção de uma ontologia de representação do conhecimento na área CS, onde além de se basear igualmente no trabalho CC2001, se propõem extensões e melhorias de forma a suportar a sua utilização num ambiente distribuído. O trabalho também estuda a possibilidade da utilização de OWL [67] como linguagem de suporte à ontologia, devido à sua referência na *Framework Semantic Web* [68][69] como um formato capaz de ligar a semântica da *Description Logic* com a liberdade sintáctica do RDF (*Resource Description Framework*) [67].

Em concreto, entre outros aspectos deste trabalho, foi feita uma primeira análise do CC2001 e decidido que se pretendia para esta ontologia um foco no apêndice A, ou seja, no corpo de conhecimento especificamente. Esta foi seguida de uma análise detalhada aos elementos do BoK, *areas*, *units*, *topics*, e particularmente *learning objectives*, também levou à descoberta de 4 tipos de relacionamentos ao nível semântico entre os vários conceitos do apêndice A:

i. ***subItemOf***
   Uma relação que essencialmente mapeia a semântica da hierarquia do apêndice A (*area* → *unit* → *topic*).

ii. ***isSuggestedFor***
Relação que sugere quais poderão ser os pré-requisitos para o estudo de determinado assunto. Essencialmente pode-se dizer que "é também sugerido conhecimento de / sobre" para a aprendizagem deste.

iii. ***isRequiredFor***
Relação que sugere especificamente pré-requisitos sobre determinado assunto. Essencialmente pode-se dizer que "é requerido saber sobre".

iv. ***isDetailedBy***
Relação que indica quais os temas ou assuntos que aprofundam o referido. Essencialmente pode-se dizer que "é detalhado em maior pormenor por".

Todas estas relações, excepto *subItemOf*, foram analisadas no actual BoK como podendo existir aplicadas em qualquer elemento (*area*, *unit*, *topic*) e a qualquer elemento, indiscriminadamente.

Tendo por base esta análise do BoK C2001, iniciou-se o trabalho de construção do modelo conceptual da ontologia, que em vez de ser construída de raiz, se baseou nos esforços do trabalho da universidade de Trento [61], o 1º anteriormente referido. Tendo sido só obtida a versão DAML+OIL do resultado do trabalho de Trento, foi necessário efectuar um processo de *reverse engineering* para se obter o modelo conceptual, que posteriormente se verificou não estar preparado para a inclusão das 4 relações acima identificadas. Foi assim proposta uma reestruturação do modelo de suporte, com a introdução de novos conceitos, conforme apresentado na Figura 2.22.

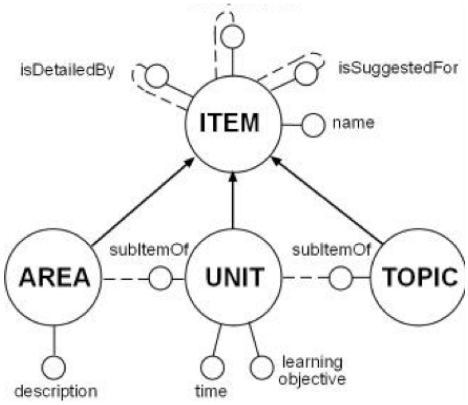


Figura 2.22 – Modelo reestruturado proposto pelo trabalho de Milão, para representação dos conceitos do apêndice A do CC2001

As alterações propostas ao modelo resumem-se à criação de um elemento genérico *item*, do qual os outros elementos descendem, tendo disponível assim as suas características e relações. Em concreto, ao nível prático da definição da

ontologia, significa que os elementos *area*, *unit* e *topic* são representados como subclasses da classe *item*, distintas entre si, e cuja sua união representará uma classe equivalente a *item*. O facto de *item* ter associado as relações anteriormente referidas (e apresentadas na figura), proporciona uma utilização das mesmas em qualquer tipo elemento (filho de *item*, portanto *area*, *unit* e *topic*), ligando-se a qualquer outro tipo de elemento. Naturalmente, de forma a ser garantida a hierarquia proposta no CC2001, a relação *subItemOf* só se aplica aos elementos específicos e na "direcção" correcta.

Em paralelo com os esforços de modelação, foi efectuado um trabalho extenso de análise entre o corpo de conhecimento CC2001 e o domínio *Computer Science*, para extrapolação de todas as relações *isSuggestedFor*, *isRequiredFor* e *isDetailedBy* possíveis entre os elementos do BoK CC2001. Este trabalho provou-se complicado, pois para a sua utilização na ontologia, este tinha de ser universalmente aceite entre todos os seus intervenientes. Para minimizar este problema, efectuou-se uma primeira extracção / extrapolação das relações entre *areas* e *units*, com base nos textos dos *learning objectives*, que foi posteriormente validada por uma comissão de elementos (da área CS) intervenientes no projecto, cada um focando-se na sua área específica. Isto produziu uma "teia" de relações semânticas com mais de 1000 ligações. Naturalmente, este esforço resultou numa ontologia modelada em OWL, posteriormente aplicada e integrada na ontologia foco principal do projecto, a ontologia SCORM (*Sharable Content Object Reference Model*) [70], mas que foge do âmbito desta dissertação.

Um posterior contacto a um dos autores deste trabalho, a Dr.ª *Laura Anna Ripamonti*, e à qual se agradece a disponibilidade, permitiu a obtenção do ficheiro OWL com a ontologia CC2001 modelada no formato apresentado, contendo as respectivas relações obtidas do trabalho de análise acima referido.

2.3.3 "*The Computing Ontology Project*"

Conforme se pôde verificar nos trabalhos das universidades italianas, a maioria dos trabalhos apresentados aplica os conceitos ACM (e relacionados) com o objectivo de estruturar ontologias. No entanto, apesar do foco estar relacionado com assunto ontologias devido ao projecto onde estão inseridos, estes trabalhos fornecem algumas sugestões para modelos bases que reforçam a necessidade da existência de algum modelo computável. Complementarmente, há outros trabalhos que se focam especificamente na construção de ontologias para a área CS, tendo uma abordagem que dá mais relevância ao estabelecimento de uma linguagem comum do que um modelo computável específico.

São de referir em particular os trabalhos desenvolvidos pela Dr.ª *Lillian N. Cassel*, que é um dos elementos envolvidos nos trabalhos ACM[11] (pertence ao

[11] Fonte, página *Lillian N. Cassel*: http://what.csc.villanova.edu/~cassel/

ACM *Education Board*, ao ACM SIGSEC – *Special Interest Group on Computer Science Education*, etc.). Além de vários trabalhos inter-relacionados, talvez os mais relevantes sejam os que abordam especificamente o projecto conhecido como "*The Computing Ontology Project*" (ou só "Ontology Project") [78] [79] [80] e o que estabelece a possibilidade da sua utilização para a definição curricular "*Using a Computing Ontology as a Foundation for Curriculum Development*" [81].

O "Ontology Project" é essencialmente um esforço complementar aos trabalhos ACM, no sentido de tentar representar a totalidade dos conceitos da área da computação e das tecnologias de informação, descrevendo de forma detalhada os vários tópicos e sub-tópicos que sejam do interesse de educadores, investigadores, profissionais, alunos, etc., permitindo a identificação de diferenças e complementaridade nas disciplinas que abordam este tópicos. Pretende-se que, além da definição da ontologia, seja também disponibilizado um mecanismo para actualização mais simplificada e regular desta informação. Na tentativa de abrangência, consideraram-se 2 tipos de fontes de informação para este projecto, sobre os tópicos e sobre objectivos, competências a adquirir, etc. As fontes utilizadas para a recolha de tópicos são essencialmente os trabalhos ACM, quer o ACM-CCS'98 [27], quer as recomendações dos volumes Computing Curricula [29][31], bem como a *Australian Computer Society* (ACS)[12] e o *German Accreditation for Informatics Program* (ASIIN)[13]. Para a recolha de objectivos e competências, consideraram-se várias universidades e organizações, como por exemplo, a *British Computer Society*[14], a Villanova University[15], o *Department of Computer Science and Engineering* da *University of Melbourne*[16], etc.

A construção da ontologia tem sido um projecto contínuo, ainda a decorrer em 2009, estando identificados 18 conceitos principais ou de topo (entenda-se da hierarquia de conceitos), conforme representados na Tabela 2.1.

Tabela 2.1 – Os conceitos de topo da *Computing Ontology*

AlgorithmsComplexity	*ComputerHardwareOrganization*	*ComputingAndNetworkSystems*
ComputingEducation	*ConceptualModeling*	*DiscreteStructures*
EthicalSocial	*GraphicsVisualizationMultimedia*	*HistoryComputing*
InformationTopics	*IntelligentSystems*	*MathematicalConnections*
ProgrammingFundamentals	*ProgrammingLanguages*	*SecurityTopics*
SystemsDevelopment	*SysemsAndProjectManagment*	*UserInterface*

A ontologia tem sido paralelamente codificada sobre a forma de um ficheiro OWL [67], que não estando ainda terminado, representa uma estrutura hierárquica em árvore e estabelece várias relações entre os conceitos. Esta

[12] http://www.acs.org.au/

[13] http://www.asiin.de/english/newdesign/index_ex5.html

[14] http://www.bcs.org/

[15] http://csc.villanova.edu

[16] http://www.cs.mu.oz.au/

ontologia pretende abranger um grande nível de detalhe da área da computação, estando definida uma árvore com 7 níveis hierárquicos: *L1-Areas*, *L2-SubAreas*, *L3-Topics*, *L4-SubTopics*, *L5-SubSubTopics*, *L6-SubSubSubTopics* e *L7-4xSubTopics*. Existem ainda alguns tipos de relações entre os vários conceitos, *IsPartOf*, *UsedBy*, etc., que tipicamente são aplicados aos níveis mais baixos da árvore (L6 e L7). Como exemplo da sua possível utilização é apresentado na Figura 2.23 um mapa das relações conceptuais existentes de e para o tópico "Testing".

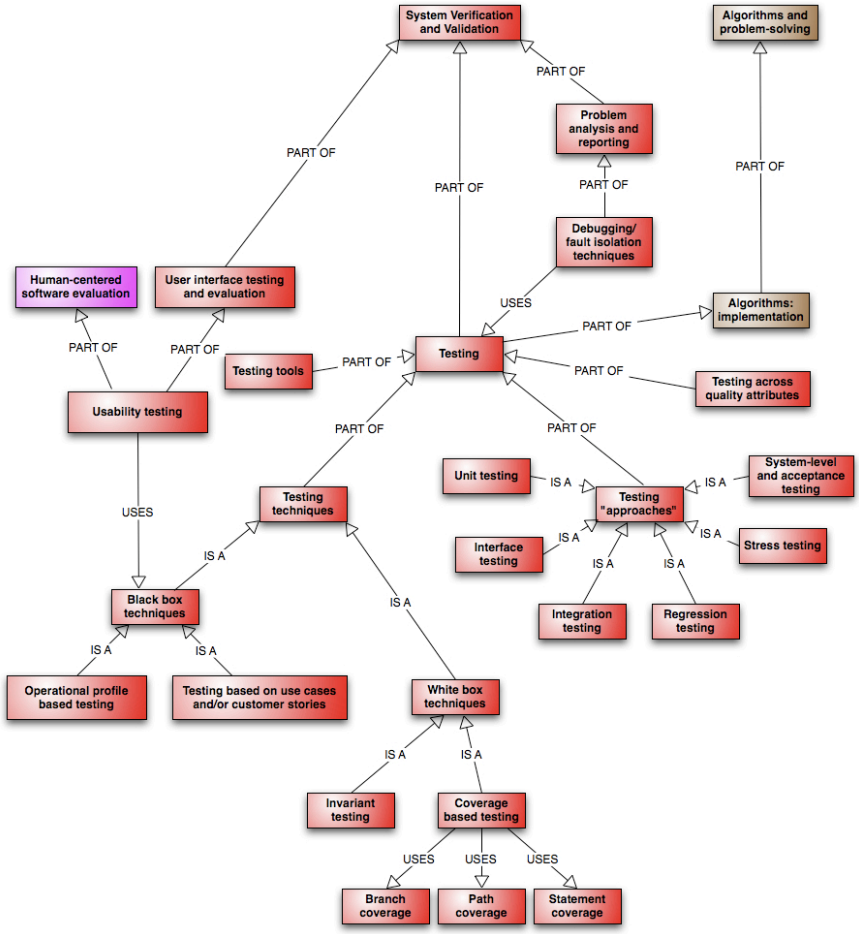


Figura 2.23 – Exemplo da utilização da ontologia – relações do tópico "TESTING"

Relativamente ao trabalho "*Using a Computing Ontology as a Foundation for Curriculum Development*", este foca alguns aspectos relevantes como o facto da ontologia não especificar qualquer tipo de requisitos curriculares, sendo efectivamente uma ferramenta para determinação do quão fundo se deverá abordar determinado assunto em determinado contexto particular. Além disso, pela participação dos autores nos trabalhos ACM, é apresentada uma visão da mudança de paradigma da orientação aos requisitos fixos das recomendações curriculares para uma orientação centrada nas características dos alunos graduados (ou seja, competências). É também proposta uma nova abordagem à definição dos *Computing Curricula*, centrada na discussão e numa ontologia

abrangente (fazendo recursos de ferramentas *Wiki* e fóruns de discussão colaborativos), com objectivo de ser mais facilmente mantida, e que se espera vir a fazer parte da nova iteração de recomendações curriculares.

Mais interessante ainda é apresentação de uma outra forma de utilização da ontologia no apoio à estruturação curricular, essencialmente baseado nos objectivos de aprendizagem / competências a adquirir e a sua relação com os tópicos da ontologia. É apresentado um cenário em que existe a possibilidade da escolha de um determinado objectivo de aprendizagem pretendido e um eventual sistema apresentar todas as ligações para conceitos relevantes e intervenientes nesse objectivo. Naturalmente estabelece-se a possibilidade deste conceito ser aplicado para um conjunto de objectivos de aprendizagem e competências, permitindo assim, com recurso à ontologia, ter uma visão estruturante dos conceitos eventualmente envolvidos e suas relações. Inclusive é sugerida a possibilidade da comparação de cursos, tendo por base os diferentes objectivos e subsequente mapa de conceitos produzido.

Por fim, pode-se dizer que devido à sua completude, é tendência imediata assumir que esta ontologia poderia ser utilizada como *Body of Knowledge* para eventual modelo proposto. No entanto, é importante referir que o trabalho deste projecto está em curso há alguns anos, não existindo quer agora, quer no período onde se efectuou pesquisa sobre eventual modelo e dados para o *Body of Knowledge*, uma versão completa e estável da mesma. Veja-se por exemplo, que os objectivos de aprendizagem referidos acima, ainda não estão à data (Agosto de 2009) mapeados no ficheiro OWL. Além disso, é importante mencionar que o próprio modelo para o *Body Of Knowledge* proposto pelo CC2001, apesar de ser anterior e mais simples, e de não ter pretensões de ser tão abrangente (por exemplo, não inclui relações entre os conceitos, algo que os trabalhos das universidades italianas propõem resolver), já estabelece esta relação dos objectivos de aprendizagem aos tópicos (conforme o BoK do apêndice A do CC2001), bem como algumas relações implícitas entre tópicos pelo facto de serem declarados em conjunto nas disciplinas da lista *course descriptions* (apêndice B do CC2001).

2.3.4 TRUC - Testable, Reusable Units of Cognition

Além destes trabalhos directamente relacionados com a utilização ou exploração dos conceitos ACM, é também importante referir outras aproximações que de alguma forma referem os trabalhos ACM, mas que não utilizem necessariamente os seus conceitos directamente. Exemplos concretos são os trabalhos desenvolvidos no *Department of Computer Science* da *Swiss Federal Institute of Technology* de Zurique (*ETH Zurich*), em particular o trabalho "*Testable, Reusable Units of Cognition*" (também referido como *TRUC*) [82] e o "*A Framework for Describing and Comparing Courses and Curricula*" [83].

No 1º trabalho é descrito um aspecto das abordagem ACM/IEEE e dos corpos de conhecimento dos Computing Curricula, que é o facto de que os tópicos referidos não estão claramente e suficientemente definidos para, por exemplo, garantir que determinado livro sobre determinado assunto cobre efectivamente e totalmente determinado tópico ACM. Sugere-se inclusive que os tópicos ACM sem um detalhe maior são apenas nomes, e como possível melhoria introduz-se o conceito de TRUC - "*Testable, Reusable Units of Cognition*". Um TRUC representa-se essencialmente por um conjunto (ou unidade) de informação detalhada, conforme a descrição e representação da Figura 2.24 (mais detalhe pode ser verificado no trabalho).

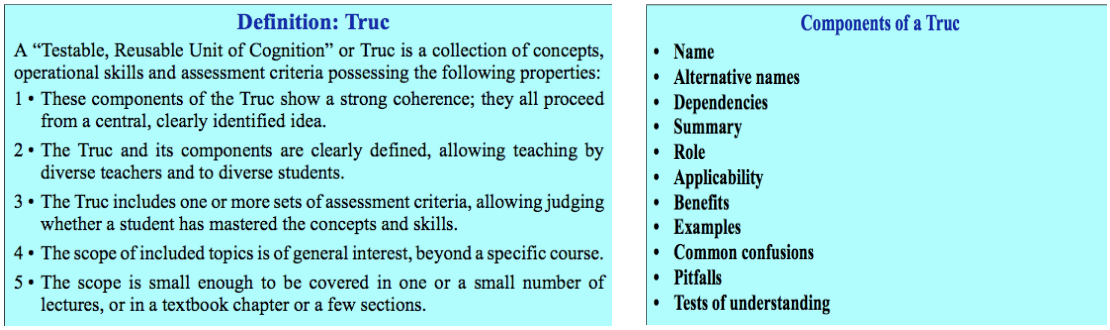


Figura 2.24 – Definição do conceito TRUC

Este conceito de TRUC poderá ainda ser agrupado em conjuntos ou estruturas hierárquicas (com os TRUC nas folhas) para representação abrangente de assuntos com vários componentes.

O 2º trabalho estende este conceito de TRUC, fazendo-lhe um conjunto de melhorias e demonstra a sua possível utilização para a eventual representação de conhecimento e definições curriculares. Essencialmente são identificadas vantagens e problemas da modelação de um corpo de conhecimento com recurso aos TRUC. Ao nível das vantagens, identifica-se por exemplo, que estrutura dos TRUC poderá facilitar a criação do mapa de tópicos e relações do corpo de conhecimento (devido à sua descrição explícita e identificação de dependências), bem como facilmente evitar ambiguidades entre os mesmos (devido à identificação de exemplos e "confusões comuns"). A aplicação dos TRUC poderá ser feita a tópicos de alto nível, o que permite por exemplo efectuar comparações entre cursos, livros, etc., mas no entanto verificaram-se alguns problemas, particularmente ao nível do detalhe. Essencialmente, na tentativa de um eventual mapeamento do corpo de conhecimento identificaram-se várias dificuldades, como a definição do nível de granularidade aplicável, formas de estabelecer os vários tipos de relações necessárias, e como recolher outros aspectos particulares do domínio do conhecimento.

Como forma de resolução destas dificuldades, e para que se possa aplicar estes conceitos a este tipo de problemas, é proposto no trabalho um conjunto de extensões do conceito TRUC simples. Em concreto introduziu-se um outro nível abaixo de TRUC, referido como *notion*, a possibilidade de relações entre *notions*

("is a" e "required", que naturalmente se estendem também para os TRUC) e a possibilidade da definição de extras TRUC, *notions* e relações, que sejam específicos do domínio em questão. Apresenta-se na Figura 2.25 um exemplo da aplicação dos TRUC e respectivas extensões, na modelação de tópicos do corpo de conhecimento sobre linguagens de programação, com uma particularidade relativa ao domínio da linguagem *Eiffel* [84] [85].

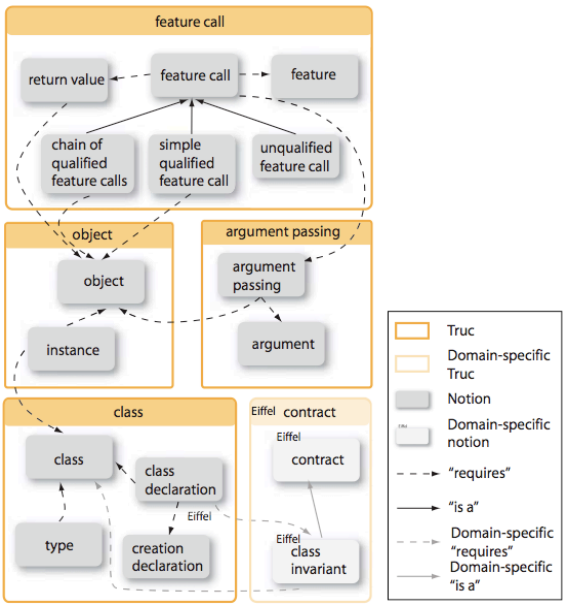


Figura 2.25 – Grafo exemplo de *notions* e TRUC

Tendo a possibilidade de modelar o corpo de conhecimento, o trabalho refere que um *curriculum* deverá ser entendido como um conjunto de TRUC (que, relembra-se, são compostos de *notions*) e as disciplinas como um conjunto de *notions* ensinadas em sequência uma após a outra. Assim, efectuar comparações entre *curriculum* ou disciplinas é comparar as suas *notions* respectivas. Como um exemplo desta possível utilização, foi efectuado um caso de estudo da cobertura de 2 livros sobre programação (um exemplo similar a disciplinas), um específico da linguagem *Eiffel* e outro da linguagem *Java* [86]. De forma a comparar ambos os livros, primeiramente foram definidos os TRUC, *notions* e relações envolvidos (estabelecendo um grafo similar ao da Figura 2.25), seguido de um mapear do conteúdo dos livros nas *notions* e TRUC respectivos. Como resultado desta comparação, produziu-se o gráfico representado na Figura 2.26, onde se apresentam o conjunto de *notions* e TRUC (anteriormente referidos) para os primeiros 3 capítulos dos livros (por uma questão de brevidade).

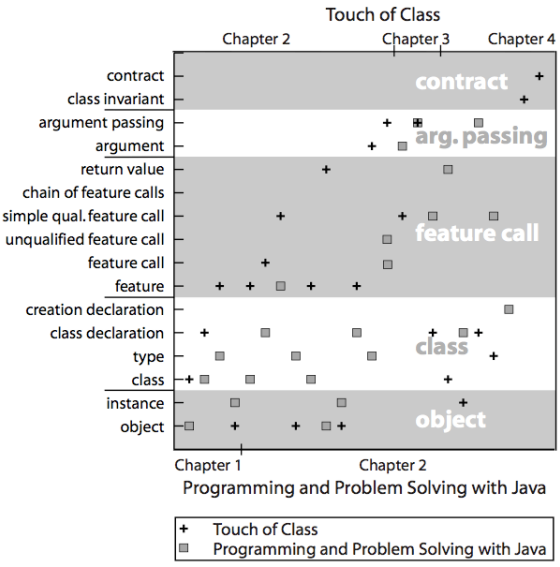


Figura 2.26 – Comparação parcial da cobertura de 2 livros

Descrevendo o gráfico, os livros estão no eixo X superior (Eiffel) e inferior (Java) com indicação sequencial dos seus capítulos, enquanto que os *notions* com o respectivo agrupamento por TRUC (em sombreado) estão no eixo Y. Cada ponto do gráfico, identificado como uma cruz ou um quadrado, representa que o *notion* ao nível do eixo Y foi abordado, respectivamente, no livro de Eiffel ou Java, no capítulo identificado no eixo X superior ou inferior correspondente. Uma análise do gráfico (a qual não se apresenta aqui directamente) permitiu ao trabalho concluir que este tipo de abordagem à utilização de TRUC e *notions*, além de facilitar uma visualização da cobertura temática dos livros, facilita também o efectuar de uma comparação entre os mesmos, da qual é possível identificar tópicos comuns, tópicos distintos e também tópicos em falta.

Concluindo que o mesmo tipo de raciocínio se pode aplicar à definição temática de cursos e disciplinas, o trabalho propõe ainda uma ferramenta de nome *TrucStudio* [83] [87] que permite a criação, manutenção e aplicação dos TRUC. Esta ferramenta permite a definição de *clusters* de TRUC, de TRUC e respectivas *notions*, bem como a definição de objectos que utilizem o corpo de conhecimento representado pelo grafo de *notions* e TRUC (cursos, disciplinas, livros, etc.). A ferramenta permite a verificação das dependências dos *notions* desses objectos e também a exportação do grafo respectivo. De futuro prevê-se a introdução de um conjunto de novas características, entre as quais o suporte para tipos de ficheiros de ontologias, o que apesar de não ser detalhado, se entende como uma forma de facilitar a definição de TRUC e *notions*, utilizando os conceitos referidos na ontologia. Note-se que a ferramenta foi desenvolvida originalmente em 2007, tendo sido evoluída ocasionalmente, estando actualmente ainda *beta* (0.2.2) [87].

Por fim, o trabalho refere a possibilidade de utilização do corpo de conhecimento ACM para a definição de TRUC e respectivos *notions*, tendo por base o trabalho *Ontology Project* e respectiva ontologia ACM [80]. São

apresentadas também algumas críticas ao facto do trabalho ACM CC2201 não estabelecer formas de verificar a compatibilidade da definição de disciplinas com as definições curriculares do respectivo apêndice B. É indicado que dentro da *ETH Zurich* se pretende evoluir para uma modelação de todas as disciplinas via TRUC, estabelecendo de futuro um repositório livremente acessível desta informação. Refere-se também que esta *Framework* poderá ser do interesse futuro do ACM, e eventualmente necessária a sua consideração em trabalhos futuros.

2.4 Trabalhos relacionados com os conceitos de Bolonha

Um dos aspectos interessantes introduzidos pelo processo de Bolonha, conforme já mencionado, refere-se à orientação do paradigma dos cursos para a aquisição de competências. Um busca na actual página do processo de Bolonha [3], mais especificamente na secção dos documentos principais de suporte, permite-nos encontrar um documento com o nome "*A Framework for Qualifications in the European Higher Education Area*" [6], onde estão convencionados para o espaço europeu de ensino superior, quais as competências gerais que os frequentadores dos 1º, 2º e 3º ciclo deverão atingir para que obtenham o grau respectivo. Esta *Framework* foi aprovada em 2005 pelos ministros europeus responsáveis pelo ensino superior, estando os mesmos comprometidos à elaboração, até 2010, de *frameworks* nacionais específicas e compatíveis com esta.

Este tema das competências é transversal às preocupações de estruturação curricular. Por exemplo, o facto de ser também introduzido pelos relatórios ACM/IEEE, demonstra uma preocupação sobre o assunto do ensino superior numa escala mais global e não apenas europeia. Em concreto, no relatório CC2001 (note-se, em 2001, antes da adopção completa do processo de Bolonha na Europa), além dos extensos *learning objectives* do apêndice A, são também referidas no capítulo 11 as capacidades e *skills* que se pretende que os alunos obtenham, bem como formas de validar essa obtenção.

Esta preocupação acerca das competências é um assunto que deverá estar também presente nos aspectos de procura de um modelo computável para a definição de estruturas curriculares. Assim, nesta secção abordam-se alguns relatórios e trabalhos que foram investigados no âmbito deste tema.

Em particular, o mais relevante é efectivamente o relatório de adequação do curso de Licenciatura em Engenharia Informática da FCT/UNL a Bolonha – "Proposta de curso de 1º ciclo de Licenciatura em Engenharia Informática" [70], pois efectuada uma abordagem extensa ao assunto.

2.4.1 Os descritores de Dublin

Também referidos ao longo deste documento na sua tradução em inglês, os *Dublin Descriptores* (DDs), são um trabalho produzido pela "*Joint Quality*

Initiative"[17], que é uma rede informal para o controle e validação da qualidade e da acreditação dos programas do 1º e 2º ciclo na Europa. Esta tem a sua origem no processo da declaração de Bolonha e foi criada numa reunião em Maastricht, onde vários países introduziram a problemática da acreditação dos seus cursos (1º e 2º ciclo).

Estes *Dublin Descriptores*, que têm o seu nome derivado da cidade onde foram inicialmente definidos em 2002, tiveram uma revisão em 2004 para a inclusão igualmente do 3º ciclo, estando consubstanciada no documento "*Shared 'Dublin' descriptors for Short Cycle, First Cycle, Second Cycle and Third Cycle Awards*" [71]. Em concreto, estes são referidos com descritores gerais, pois pretendem estipular uma classificação genérica e partilhável, acerca de competências académicas, elas também genéricas, que um aluno deverá obter para se considerar graduado de determinado ciclo em determinado curso. O objectivo foi criar uma classificação aplicável genericamente às competências fornecidas por vários cursos em várias universidades, estabelecendo assim um eventual critério global aceitável. Após uma análise do documento DDs de 2004 [72], onde se identificam competências específicas para os 3 ciclos, verifica-se que são exactamente as mesmas da *Framework* das competências propostas de Bolonha [6] (atrás referida), ficando entendido que os DDs foram adoptados oficialmente pelo processo de Bolonha. Note-se por exemplo, que são também equivalentes aos requisitos em Portugal para a obtenção do grau de licenciado, mestre e doutor, conforme descritos nos artigos 5º, 15º e 28º do Decreto-Lei nº 74/2006 [20].

Além de um detalhe das competências requeridas para os vários ciclos, os DDs genéricos podem ser resumidos como uma classificação de 5 tipos (ou descritores), que corresponderão a diferentes competências que aumentarão complexidade entre cada ciclo:

1. *Knowledge and understanding;*
2. *Applying knowledge and understanding;*
3. *Making judgements and decision making;*
4. *Communication;*
5. *Learning and self-learning skills.*

O documento DDs, estabelece em detalhe as diferenças de competências de cada descritor entre cada ciclo, conforme se pode verificar na Figura 2.27.

[17] http://www.jointquality.nl/

Differentiating between cycles

Cycle	Knowledge and understanding:
1 (Bachelor)	[Is] supported by advanced text books [with] some aspects informed by knowledge at the forefront of their field of study ..
2 (Master)	provides a basis or opportunity for originality in developing or applying ideas often in a research* context ..
3 (Doctorate)	[includes] a systematic understanding of their field of study and mastery of the methods of research* associated with that field..

Figura 2.27 – As competências de cada ciclo para o *Dublin Descriptor* do tipo 1

2.4.2 Descritores de Dublin específicos – Holanda (Engenharia)

Além dos *Dublin Descriptores*, e seguindo a provocação feita em Bolonha aos ministros para a elaboração de *frameworks* nacionais, alguns países efectuam ou já tinham efectuado trabalhos neste sentido. Em particular na Holanda, existe um trabalho de 2005, desenvolvido por um conjunto de universidades, *Delft University of Technology*, *Eindhoven University of Technology* e *University of Twente*, intitulado "*Criteria for Academic Bachelor's and Master's Curricula*" [73], que estabelece um conjunto de descritores mais focados na área de engenharia, especificamente para o 1º e 2º ciclo de Bolonha.

Este trabalho, que em diante será referido como descritores de Dublin para engenharia, ou *Dublin Descriptors – Engineering* (e abreviado como DDs-e), define 7 áreas de competência que caracterizam um aluno (e que estão descritas em detalhe no documento [73]):

1. *Competent in one or more scientific disciplines*
2. *Competent in doing research*
3. *Competent in designing*
4. *A scientific approach*
5. *Basic intellectual skills*
6. *Competent in co-operating and communicating*
7. *Takes account of the temporal and social context*

O documento apresenta ainda relações entre as áreas referidas, conforme a Figura 2.28 e a seguinte descrição: a) o domínio do conhecimento que é estudado (competências 1, 2 e 3); b) os métodos académicos de pensar e agir (competências 4,5 e 6); c) o contexto onde se pratica (competências 7).

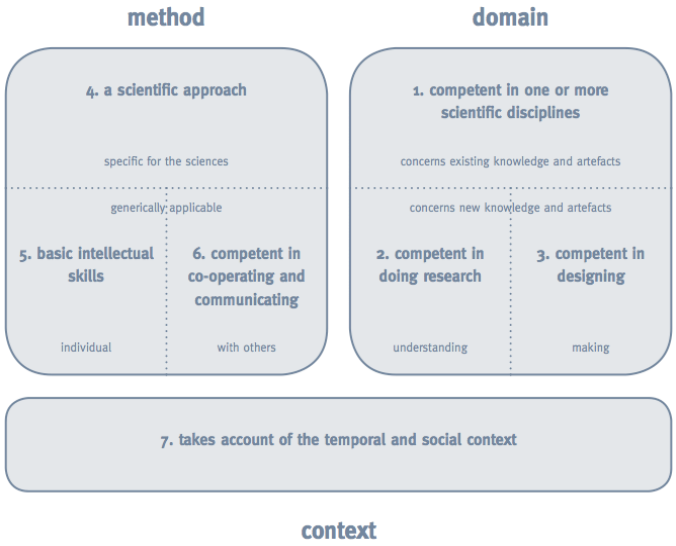


Figura 2.28 – As 7 áreas de competência e suas relações, propostas pelo documento das universidades Holandesas (fonte [73])

Um pormenor muito interessante neste trabalho é que cada área de competência, ou DD-e, é posteriormente detalhado em maior pormenor com várias frases, com uma distinção específica entre as competências de um graduado do 1º ciclo e do 2º ciclo. Além disso, para evitar ambiguidades, **é ainda proposto em cada frase de detalhe da competência no respectivo ciclo, uma indicação de qual o foco (ou focos) da competência**, sobre Conhecimento (*Knowledge* – K), Capacidade (*Skill* – S) ou Atitude (*Attitude* – A). Na análise efectuada no documento, indica-se que para o 1º ciclo o conhecimento e capacidade estão quase sempre combinados, enquanto que para o 2º ciclo se foca mais num aspecto de atitude (Como exemplo, é referido que não é suficiente para um "mestre" saber, ou saber fazer alguma coisa, é necessário que tenha também uma atitude para a utilização desse conhecimento ou capacidade em situações relevantes). Veja-se o exemplo da área 5 - *Basic intellectual skills* na seguinte Figura 2.29.

Referindo o que se descreveu acima, veja-se a 3ª linha da tabela da Figura 2.29, e particularmente as letras entre parêntesis rectos no final das frases. Nesta refere-se que para o 1º ciclo espera-se que o graduado tenha conhecimento sobre os vários modos de raciocínio utilizados na área, portanto uma competência focada em conhecimento e capacidade (veja-se o K e S), enquanto que para um graduado do 2º ciclo espera-se que sabia aplicar esses mesmos modos, portanto uma competência focada na atitude e na aplicação do conhecimento e capacidade obtida (veja-se o K, S, A).

Por fim, além de se abordar genericamente uma possível forma de atribuição de valores e níveis às diferentes competências (em concreto sugere-se a utilização de dimensões: *analytic*, *synthetic*, *abstract* e *concrete*), especificam-se várias formas possíveis de utilização destas competências: *i*) para a definição de objectivos de aprendizagem gerais dos currículos universitários ou de cursos especificamente; *ii*) para a descrição, comparação, análise e avaliação de cursos;

iii) para testar as competências académicas dos alunos; *iv*) e para a indicação da cobertura das competências de cursos (chama-lhe especificamente "perfil académico").

Bachelor	Master
Is able (with supervision) to critically reflect on his or her own thinking, decision making, and acting and to adjust these on the basis of this reflection. [ks]	Idem, independently. [ksa]
Is able to reason logically within the field and beyond; both 'why' and 'what-if' reasoning. [ks]	Is able to recognise fallacies. [ks]
Is able to recognise modes of reasoning (induction, deduction, analogy etc.) within the field. [ks]	Is able to apply these modes of reasoning. [ksa]
Is able to ask adequate questions, and has a critical yet constructive attitude towards analysing and solving simple problems in the field. [ks]	Idem, for more complex (real-life) problems. [ksa]
Is able to form a well-reasoned opinion in the case of incomplete or irrelevant data. [ks]	Idem, taking account of the way in which that data came into being. [ks]
Is able to take a standpoint with regard to a scientific argument in the field. [ksa]	Idem, and is able to assess this critically as to its value. [ksa]
Possesses basic numerical skills, and has an understanding of orders of magnitude. [ks]	Idem. [ksa]

Figura 2.29 – Competências detalhe da área 5. *Basic intellectual skills* (fonte [73])

Um exemplo interessante desta última utilização é apresentado como um gráfico na seguinte Figura 2.30, onde para um curso fictício, se efectua uma distribuição, também fictícia, de ECTS para cada área de competência. Este define o perfil académico de competências desse curso fictício num formato radial, que pode ser interessante para visualmente se tirar conclusões sobre a sua cobertura ou permitir efectuar comparações entre os perfis dos vários cursos.

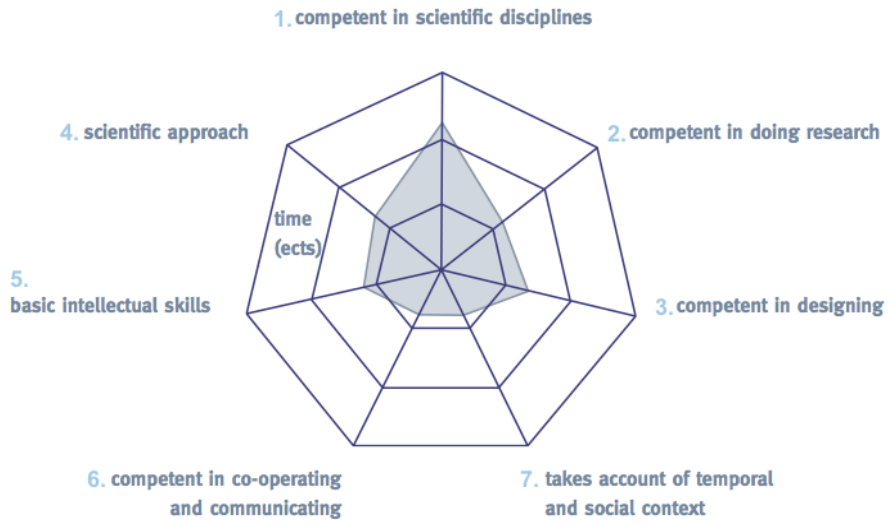


Figura 2.30 – Exemplo do perfil académico de competências de um curso (fonte [73])

2.4.3 A Taxonomia de *Bloom* e as competências

Se for efectuada uma leitura dos vários agrupamentos de competências até aqui referidos, identifica-se que tipicamente as frases contêm conjunto de verbos mais ou menos reduzido, que indiciam acções relacionadas com a aprendizagem ou capacidade de execução de alguma coisa.

Em concreto, quando se fala de verbos ou acções relacionados com capacidades e aprendizagem, está-se a centrar sobre aspectos do domínio cognitivo. Neste domínio, existem trabalhos igualmente relacionados com o nosso assunto das competências, por exemplo, e em particular, o trabalho de *Benjamin S. Bloom*, conhecido como a taxonomia de *Bloom ("Bloom's Toxonomy")* [74][75]. Este trabalho teve a sua origem em 1948, com base na pesquisa de um grupo de investigadores liderados por *Bloom* [75], cujo o principal objectivo era desenvolver um método de classificação para comportamentos intelectuais que fossem relevantes ao processo de aprendizagem. Esta pesquisa tornou-se posteriormente numa taxonomia sobre 3 domínios de actividades educacionais: **Cognitivo** (aprendizagem e competências "mentais"), **Afectivo** (aprendizagem da área emocional e social) e **Psicomotor** (competências manuais e físicas). De imediato se podem estabelecer paralelos com as 3 formas de classificação de competências atrás mencionadas, ***Knowledge*** → **Cognitivo**, ***Skills*** → **Psicomotor**, e ***Attitude*** → **Afectivo**. A pesquisa deste grupo focou-se principalmente no domínio cognitivo e afectivo, tendo estabelecido uma subdivisão dos 3 domínios que vão dos comportamentos mais simples aos mais complexos.

O trabalho de pesquisa deste grupo especificamente sobre o domínio cognitivo é aquilo que é conhecido como a taxonomia de *Bloom*, podendo ser descrito como um modelo de classificação do pensamento de acordo com 6 níveis evolutivos de complexidade, conforme a representação na Figura 2.31.

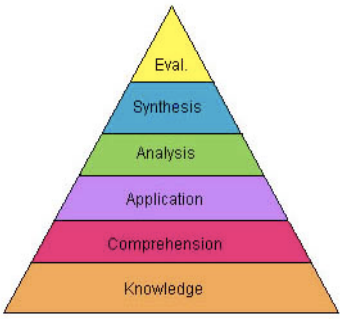


Figura 2.31 – Os 6 níveis da taxonomia de *Bloom* (fonte [75])

Muito simplificadamente, a ideia da taxonomia é que aquilo que os educadores querem que os estudantes aprendam, podem ser organizado nesta hierarquia da menor para a maior complexidade, existindo um conjunto de verbos e frases que identificam os comportamento de cada nível. Na seguinte Figura 2.32 apresentam-se um conjunto de alguns dos verbos que identificam o comportamento em cada nível.

			Critical Thinking		Evaluation
				Synthesis	Judge
			Analysis	Design	Appraise
		Application	Compare	Plan	Estimate
	Comprehension	Use	Distinguish	Compose	Evaluate
	Express	Employ	Differentiate	Propose	Revise
Knowledge	Restate	Interpret	Diagram	Formulate	Score
Define	Identify	Dramatize	Analyze	Arrange	Select
Repeat	Explain	Sketch	Categorize	Assemble	Rate
Name	Recognize	Practice	Appraise	Collect	Choose
Recall	Discuss	Illustrate	Experiment	Construct	Measure
List	Describe	Operate	Test	Create	Compare
Relate	Tell	Demonstrate	Contrast	Setup	Value
Record	Locate	Apply	Inspect	Organize	Assess
Underline	Report	Schedule	Debate	Prepare	
Outline	Review	Show	Inventory	Manage	
Delineate	Summarize	Translate	Question	Predict	
Specify		Interpret	Examine		
State		Solve	Criticize		
Label		Sketch	Relate		
Match			Solve		
			Calculate		
			Critique		
			Classify		

Figura 2.32 - Verbos da taxonomia de *Bloom* - domínio cognitivo (retirado de [18])

Especificamente na Figura 2.33, está representada uma roda baseada na taxonomia de *Bloom*, que contém no círculo mais central os 6 níveis da taxonomia, no círculo dos meio os verbos correspondentes a cada nível, e no círculo exterior alguns dos produtos típicos produzidos para avaliação do nível em questão.

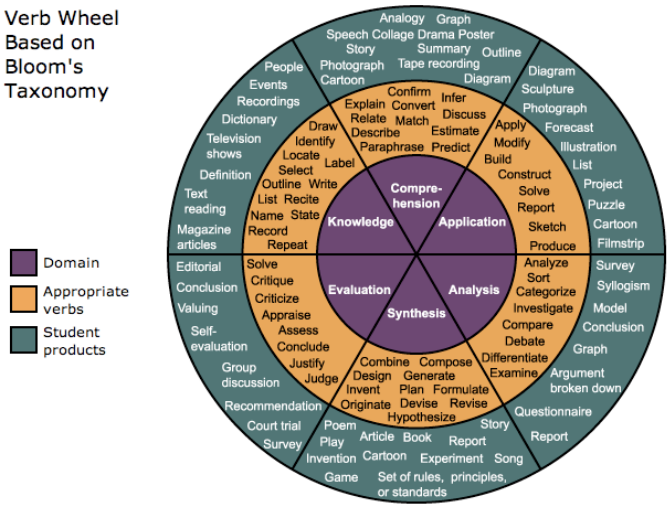


Figura 2.33 – Roda da taxonomia de Bloom (retirado de [19])

Por fim, refere-se ainda a existência de um trabalho publicado em 2001, "*Revised Bloom's Taxonomy*" [76], desenvolvido por um grupo de investigadores liderados por *Lorin Anderson* (um antigo aluno de *Bloom*) que introduziram diversas pequenas alterações [77]. Nesta dissertação, optou-se por considerar a taxonomia original pela sua adopção mais generalizada, sendo no entanto possível a adopção da taxonomia revista no futuro.

[18] Fonte: http://keep2.sjfc.edu/faculty/jpriola/560/word/bloom_domains.pdf

[19] Fonte: http://cstep.csumb.edu/Obj_tutorial/bloomwheel.html

2.5 A proposta de adequação a Bolonha do D.I. FCT/UNL

O relatório de adequação do curso de Licenciatura em Engenharia Informática da FCT/UNL a Bolonha – "Proposta de curso de 1º ciclo de Licenciatura em Engenharia Informática" [71], já anteriormente referido, é um elemento muito relevante para esta dissertação pois como estabelece as linhas orientadoras do curso 1º ciclo, fornece várias pistas e sugestões para um eventual modelo computável, bem como dados para possível alimentação e validação do mesmo.

Relativamente ao assunto desta secção, os aspectos relacionados com Bolonha, este relatório analisa e compara vários trabalhos sobre competências, que são importantes mencionar.

Além da referência aos *Dublin Descriptors* genéricos e a*os Dublin Descriptors* específicos para a área de Engenharia (atrás referidos), que de certa forma validam a relevância destes trabalhos nesta dissertação, indica-se que estes últimos se adequam aos objectivos traçados para o curso, pelo que serão adoptados para a caracterização de competências no relatório. Adicionalmente é efectuada uma correspondência entre ambos os tipos de DDs, pois conforme já referido anteriormente, os DDs originais são equivalentes aos requisitos em Portugal para a obtenção do grau de licenciado, mestre e doutor (conforme Decreto-Lei nº 74/2006 [20]), sendo útil a compatibilização com os DDs-e adoptados. Além disso, esta compatibilização dos vários DDs poderá ser poderá ser relevante para a representação de várias formas de competências num eventual modelo computável.

Tabela 2.2 – Correspondência entre *Dublin Descriptors* da área da engenharia e *Dublin Descriptors* genéricos

***Dublin Descriptors* Eng.**	***Dublin Descriptors***
A - *Competent in one or more scientific disciplines;*	*1. Knowledge and understanding* *2. Applying knowledge and understanding;*
B - *Competent in doing research;*	*2. Applying knowledge and understanding;* *3. Making judgements and decision making;* *5. Learning and self-learning skills;*
C - *Competent in designing;*	*2. Applying knowledge and understanding;* *3. Making judgements and decision making;*
D - *A scientific approach;*	*1. Knowledge and understanding* *2. Applying knowledge and understanding;* *3. Making judgements and decision making;*
E - *Basic intellectual skills;*	*1. Knowledge and understanding;* *3. Making judgements and decision making;*
F - *Competent in co-operating and communicating;*	*2. Applying knowledge and understanding;* *4. Communication;*
G - *Takes account of the temporal and social context;*	*3. Making judgements and decision making;* *5. Learning and self-learning skills;*

Na sequencia da utilização de competências para detalhe da cobertura do curso e das respectivas disciplinas no relatório, foram também identificados os possíveis agrupamentos das competências, 1) Saber, 2) Fazer e 3) *Soft-skill*, que são de certa forma similares, à classificação proposta nos DDs-e, *Knowledge*, *Skill* e *Attitude*. Estabeleceu-se também que, para competências do tipo "saber" e "fazer", dado que são específicas de cada disciplina, não se definiu nenhum quadro de referência particular (mas sugere-se o BoK ACM/IEEE do CC2001), enquanto que para as competências do tipo "*soft-skill*", se definiu um quadro de referência específico para o curso, que poderá eventualmente ser compatibilizado com os DDs.

Este quadro de referência de *soft-skills* proposto para o 1º ciclo do curso de Eng. Informática no D.I., e que se apresenta na Figura 2.34, é também um bom elemento para eventual adopção num modelo computável de definição de um curso.

Anexo D. Taxinomia de referência para *soft skills*

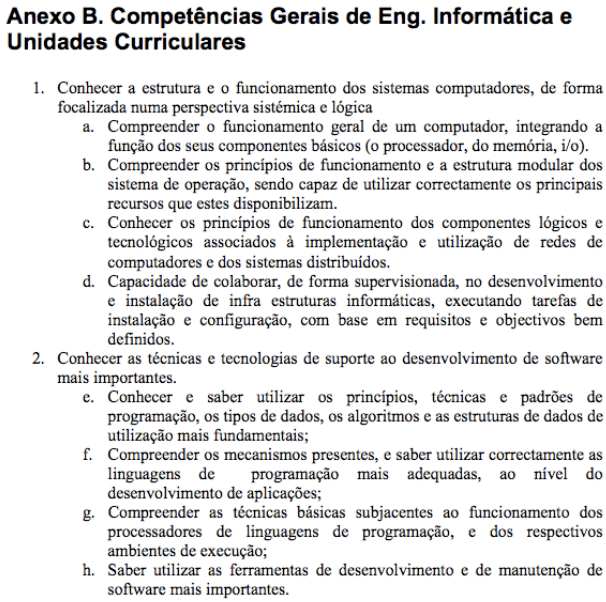
1. Geral
   - Capacidade de comunicação oral
     - Exposição oral de uma ideia, uma duvida, etc.
     - Apresentação oral com slides ou outro material multimédia
     - Realização de uma demonstração
   - Capacidade de comunicação escrita
     - Relatório da análise, desenho e implementação de uma solução
     - Relatório de um trabalho experimental
     - Resumo ou síntese de um tema
     - Relatório de uma análise (comparativa)
   - Capacidade de organizar o trabalho
   - Capacidade de gestão do tempo e cumprimento de prazos
   - Capacidade de trabalhar em equipa e de colaborar numa equipa
   - Capacidade e atitude de iniciativa
   - Capacidade e postura para pensamento critico
   - Capacidade de investigação e autonomia
   - Criatividade e Imaginação
   - Capacidade de concretização
   - Atitude de exigência e qualidade
   - Honestidade
2. Científicos
   - Cultivar o rigor científico
   - Capacidade de raciocínio abstracto e formal
   - Capacidade de modelação abstracta
   - Capacidade de conduzir e avaliar uma actividade experimental com base científica
3. Engenharia
   - Capacidade de modelação de problemas
   - Capacidade para efectuar escolhas fundamentadas
   - Capacidade para seleccionar instrumentos apropriados a um problema
   - Capacidade de avaliação crítica de uma solução
   - Reconhecer situações e problemas que requerem soluções sub-optimais

Figura 2.34 – Quadro de referência de *soft-skills*, proposto para o curso no D.I.

Adicionalmente, como forma de cumprir o requisito de adequação a Bolonha, apresentou-se também um enquadramento justificativo das competências, seguindo os *Dublin Descriptors* atrás mencionados. Este enquadramento identifica especificamente que para o 1º descritor (A. *Competent in one or more scientific disciplines*) existe a possibilidade de se demonstrar as competências com recurso a 2 tipos de quadros de referência: 1) uma matriz de competências técnicas gerais de Engenharia Informática, elaborada pela comissão de responsáveis do 1º ciclo do Departamento de Informática da FCT/UNL; e 2) uma matriz de competências específicas, baseada na classificação proposta no BoK CC2001. Em concreto, para esta tese, isto permite 2 coisas:

i. Obter mais uma lista de competências técnicas gerais (a matriz referida 1), representada parcialmente na Figura 2.35, e que poderá ser considerada para o modelo de informação computável.

ii. Validar que aplicação do BoK CC2001 (em concreto *areas* e *units*) poderá servir para modelar, pelo menos parcialmente, os temas a abordar e as competências a adquirir das disciplinas do actual 1º ciclo do curso de Eng. Informática do D.I.



Anexo B. Competências Gerais de Eng. Informática e Unidades Curriculares

1. Conhecer a estrutura e o funcionamento dos sistemas computadores, de forma focalizada numa perspectiva sistémica e lógica
   a. Compreender o funcionamento geral de um computador, integrando a função dos seus componentes básicos (o processador, do memória, i/o).
   b. Compreender os princípios de funcionamento e a estrutura modular dos sistema de operação, sendo capaz de utilizar correctamente os principais recursos que estes disponibilizam.
   c. Conhecer os princípios de funcionamento dos componentes lógicos e tecnológicos associados à implementação e utilização de redes de computadores e dos sistemas distribuídos.
   d. Capacidade de colaborar, de forma supervisionada, no desenvolvimento e instalação de infra estruturas informáticas, executando tarefas de instalação e configuração, com base em requisitos e objectivos bem definidos.
2. Conhecer as técnicas e tecnologias de suporte ao desenvolvimento de software mais importantes.
   e. Conhecer e saber utilizar os princípios, técnicas e padrões de programação, os tipos de dados, os algoritmos e as estruturas de dados de utilização mais fundamentais;
   f. Compreender os mecanismos presentes, e saber utilizar correctamente as linguagens de programação mais adequadas, ao nível do desenvolvimento de aplicações;
   g. Compreender as técnicas básicas subjacentes ao funcionamento dos processadores de linguagens de programação, e dos respectivos ambientes de execução;
   h. Saber utilizar as ferramentas de desenvolvimento e de manutenção de software mais importantes.

Figura 2.35 – Lista (parcial) de competências técnicas gerais do 1º ciclo D.I.

Por fim, nos anexos deste relatório existe bastante informação útil para eventual apoio à construção de um modelo de informação computável, particularmente no aspecto de alimentação de informação. A saber:

i. Um cruzamento entre as *areas* e *units* BoK CC2001, e as disciplinas do curso, conforme se verifica na Figura 2.36.

ii. Uma matriz de cobertura das "competências técnicas gerais" nas disciplinas do curso, conforme a Figura 2.37.

iii. Uma matriz de incidências das unidades curriculares nos Descritores de Dublin de Engenharia, conforme a Figura 2.38.

É igualmente relevante mencionar que a informação de i, ii e iii, acima referida, tem origem no trabalho manual e exaustivo, exclusivo da comissão do 1º ciclo, não sendo o resultado de uma derivação directamente computável (ou mesmo manual) da definição das disciplinas.

PF. Programming Fundamentals (38 core hours)
PF1. Fundamental programming constructs
Introdução à Programação
PF2. Algorithms and problem-solving
Introdução à Programação
Programação Orientada pelos Objectos
PF3. Fundamental data structures
Introdução à Programação
Algoritmos e Estruturas de Dados
PF4. Recursion
Algoritmos e Estruturas de Dados
PF5. Event-driven programming
Programação Orientada pelos Objectos

AL. Algorithms and Complexity (31 core hours)
AL1. Basic algorithmic analysis
Algoritmos e Estruturas de Dados
AL2. Algorithmic strategies
Análise e Desenho de Algoritmos
AL3. Fundamental computing algorithms
Introdução à Programação
Algoritmos e Estruturas de Dados
Análise e Desenho de Algoritmos

Figura 2.36 – Mapeamento das *areas* e *units* CC2001 com respectivas disciplinas que as abordam (figura parcial)

Unidade Curricular	Área	ECTS	a	b	c	d	e	f	g	h	i	j	k	4	5	6	7
Análise Matemática 1	MAT	6													x		
Introdução à Programação	INF	9	x				x	x	x	x	x		x				
Intro aos Sistemas e Redes Computadores	INF	6	x	x	x	x											
Lógica Computacional	INF	6												x	x		
Expressão e Comunicação	CHS	3												x		x	
Análise Matemática 2	MAT	6													x		
Matemática Discreta	MAT	6												x	x		
Arquitectura de Computadores	INF	6	x	x	x	x											
Programação Orientada pelos Objectos	INF	6					x	x	x	x	x	x	x				
Pensamento Crítico	CHS	6										x	x			x	
Álgebra Linear e Geometria Analítica	MAT	6													x		

Figura 2.37 - Matriz (parcial) disciplinas vs. "competências técnicas gerais"

Disciplina	Área	ECTS	A	B	C	D	E	F	G
Análise Matemática 1	MAT	7	x			x	x		
Introdução à Programação	INF	8	x		x	x			
Intr. aos Sistemas e Redes Computadores	INF	6	x				x		
Lógica Computacional	INF	6	x			x	x		
Expressão e Comunicação	CHS	3		x				x	
Análise Matemática 2	MAT	6	x			x	x		
Matemática Discreta	MAT	6	x			x	x		
Arquitectura de Computadores	INF	6	x	x			x		

Figura 2.38 – Matriz (parcial) disciplinas vs. DDs-eng.

2.6 Algumas conclusões sobres os trabalhos apresentados

Tendo uma visão detalhada dos vários tipos de trabalhos investigados no âmbito da dissertação, é importante apresentar algumas conclusões imediatas à sua aplicabilidade e utilização na dissertação.

O trabalho ACM-CCS foca-se num sistema de classificação, que se pode entender também como uma primeira tentativa de aproximação a uma taxonomia da área CS. O seu principal objectivo não está direccionado para o estabelecimento um corpo de conhecimento, e sim para a classificação de livros,

trabalhos, etc., pelo que no âmbito desta dissertação considera-se a sua utilização como eventual classificação da bibliografia das disciplinas da estrutura curricular. Pelo facto deste sistema ser amplamente divulgado e bastante utilizado na classificação de livros e trabalhos académicos, poderá também auxiliar a identificação e distinção dos tópicos do corpo de conhecimento CS que são abordados em determinada peça da bibliografia, facilitando a sua integração em disciplinas, a verificação da sua cobertura e eventual comparabilidade com outros trabalhos. Naturalmente, a utilização deste sistema beneficiaria de um eventual trabalho de ligação ou cruzamento dos tópicos ACM-CCS aos tópicos do corpo de conhecimento que se eventualmente utilizará, permitindo com base nuns, inferir parcialmente os outros.

Relativamente aos trabalhos ACM/IEEE, os *computing curricula*, percebe-se pelo ênfase que lhes é dado, que são bastante relevantes para esta dissertação. Os motivos são relativamente óbvios, pois estes estabelecem nas várias subáreas dentro da área da computação, as bases para os corpos de conhecimento específicos, efectuam recomendações para estruturação curricular, propondo inclusive um conjunto base inicial de definições para as disciplinas *core* necessárias nos cursos. Mais em concreto, focou-se no trabalho CC2001, que é a recomendação relativa à subárea *Computer Science,* por vários motivos:

- É a subárea com uma maior abrangência ao nível temático e a que melhor retrata o actual 1ºciclo do curso do Departamento de Informática.
- Pelo histórico da evolução dos *computing curricula*, o CC2001 foi primeiramente desenvolvido, estando as recomendações das outras áreas baseadas neste.
- Além disso, faria sentido considerar uma recomendação que já tivesse sido "testada" ou "experimentada", requisito que o CC2001 cumpre novamente pelo facto de ter sido o primeiro. A adopção do CC2008 não foi considerada pois este só foi disponibilizado após os trabalhos mais práticos desta dissertação, só existindo actualmente sobre versão em PDF (o que torna mais complicada a extracção automatizada de informação).
- O corpo de conhecimento proposto no CC2001 está explicitamente descrito e é abrangente (mas não totalmente completo e actualizado), propondo um modelo relativamente claro e simplificado, que é genericamente utilizável em qualquer universidade e qualquer curso desta área. A simplicidade do modelo torna-o também algo extensível, permitindo a sua futura evolução.
- O facto de não serem apenas feitas recomendações curriculares, mas também serem apresentadas descrições detalhadas das principais disciplinas *core* e da sua respectiva cobertura temática, é muito útil. Isto estabelece um modelo base simplificado para a declaração de

disciplinas, focado nos aspectos de cobertura, que poderá ser utilizado e eventualmente estendido. Mas igualmente importante é o facto destas descrições serem por si só um base de conhecimento sobre um curso com uma cobertura mínima inicial, que poderá ser utilizado como base para a definição de cursos novos e para comparação com outros cursos já definidos (verificar cobertura de temas, disciplinas e assuntos mínimos necessários, etc.).

- Por fim, igualmente bastante relevante, é o facto da recomendação CC2001 existir também em formato digital, em particular sob a forma de HTML, o que facilita todo um eventual processo de extracção e utilização da informação do mesmo (em particular dos apêndices A e B).

Além do foco no CC2001, refere-se que os outros relatórios posteriores, referidos no CC2005, estabelecem definições mais detalhadas para subáreas específicas (corpo de conhecimento, competências e estrutura curricular), podendo eventualmente servir também para auxiliar a definição dos perfis dentro do 1º ciclo do curso de Eng.ª Informática.

Estes motivos apresentados, relacionam-se inclusivamente com alguns dos motivos enunciados pelos trabalhos que se referiram como utilizadores dos conceitos *computing curricula*. Em particular, os trabalhos das universidades italianas reforçam a possibilidade da extracção e utilização do corpo de conhecimento CC2001 (no caso italiano, foi inclusive feita uma extracção directa do PDF), bem como a possibilidade de extensão do seu modelo com um conjunto adicional de informação. O modelo e extensão proposta por estes trabalhos, bem como os dados recolhidos para a sua corporização, são de bastante interesse para esta dissertação. Estes estabelecem um conjunto de informação adicional, que estende o corpo de conhecimento base CC2001 e que permitirá ter outro tipo de relações entre as *areas*, *units* e *topics*, encontrando assim novas formas de apoiar a definição curricular e de verificação de cobertura curricular. Considerou-se então nesta dissertação a adopção de alguns componentes desta proposta de extensão do modelo, que permitirão a definição de um modelo mais completo e um corpo de conhecimento "aumentado", apoiando assim a construção de um protótipo de validação do modelo.

Na sequência dos anteriores, os trabalhos relacionados com o *Ontology Project* utilizam os *computing curricula* para estabelecerem a possibilidade da evolução do corpo de conhecimento para uma ontologia específica da área. Estes trabalhos foram úteis para esta dissertação por 2 motivos: além de voltarem a reforçar a possibilidade de utilização e extracção da informação do corpo de conhecimento ACM, permitiram apoiar a conclusão que os trabalhos da dissertação para obtenção de um modelo computável não passam necessariamente por problemas de modelação de ontologias, mas sim por um foco concreto na construção de um modelo. Inicialmente, o esforço desenvolvido nestes trabalhos para extracção e

modelação da ontologia, levou a conclusões que se consideraram relevantes para a dissertação:

i. verificou-se que corpo de conhecimento ACM beneficiaria de relações adicionais entre os seus elementos e que poderia ser ainda detalhado em mais níveis abaixo dos actuais tópicos;

ii. verificou-se que se poderiam utilizar as relações a partir de determinado ponto ou conceito do corpo de conhecimento (ou da ontologia), estabelecendo assim "mapas" de conhecimento relacionado, eventualmente utilizável para comparações e outros fins;

iii. por fim, verificou-se também que é efectivamente possível transpor o corpo de conhecimento para uma ontologia, sendo possível uma representação da mesma no formato OWL.

Posteriormente, após uma análise destes trabalhos, confirmou-se que efectivamente a representação do corpo de conhecimento é possível ser feita com recurso a diversos tipos de tecnologias e formatos, sendo a ontologia apenas uma forma entre várias. **Percebeu-se também que os problemas relacionados com a criação e modelação de ontologias, não terão efectivamente grande impacto no problema que a dissertação pretende primeiramente abordar**. O foco não deverá ser especificamente nos aspectos técnicos do formato de representação do conhecimento, **mas sim na procura e definição de um modelo de conhecimento consistente**, que permita apoiar a prova da utilidade de um modelo computável. Alguns aspectos mais técnicos da representação do modelo proposto na dissertação, são apresentados no Capítulo 4, onde se descreve um protótipo exemplo de suporte. Não estando o foco da dissertação no procurar da solução tecnológica ideal, é de referir que esta proposta de protótipo beneficiou naturalmente de algumas das conclusões deste trabalhos investigados, principalmente relativamente às formas e formatos utilizados para a representação destes modelos abordados.

Como contraposição aos trabalhos que fazem directamente uso dos *computing curricula* ACM, foram apresentados os trabalhos do conceito TRUC. Estes têm uma abordagem diferente à modelação do corpo de conhecimento, mais abstracta, que foi essencialmente orientada ao aumento de informação sobre cada conceito, de forma a evitar eventuais ambiguidades e interpretações erróneas (algo que pode efectivamente ocorrer com o ACM). Estes trabalhos reforçam também algumas das conclusões dos trabalho anteriores: tendo um corpo de conhecimento bem modelado e com informação suficiente, é possível efectuar operações de comparação de objectos que tenham o seu conteúdo reflectido nesse corpo, sejam cursos, disciplinas, e livros, permitindo a identificação de padrões, áreas ou tópicos não cobertos, etc. No entanto, após uma análise destes trabalhos, é possível concluir-se 3 aspectos:

i. A abordagem recomenda a introdução de novos conceitos adicionais aos TRUC, que efectivamente são correspondentes aos 3 níveis do

corpo de conhecimento ACM: *cluster of TRUC* → *area*, TRUC → *unit*, *notion* → *topic*.

ii. Estes trabalhos não produziram com esta abordagem uma modelação efectiva de um corpo de conhecimento completo (não sendo por isso subsequentemente disponibilizado livremente).

iii. As vantagens desta abordagem podem ser também reflectidas directamente no tradicional corpo de conhecimento ACM, considerando a possibilidade de introduzir informação adicional e relações extras (conforme os trabalhos italianos advogam).

Sobre os objectos que utilizarão os conceitos do corpo de conhecimento, cursos, disciplinas ou livros, não é proposto nenhum modelo concreto específico. No entanto, a ferramenta apresentada pelos trabalhos para recolha de informação de um corpo de conhecimento e para a definição de objectos que o utilizarão, indica a possibilidade de efectuar validações sobre o grafo de relações, bem como disponibilizar funcionalidades de comparação entre os objectos. Isto é interessante para a dissertação e para os seus objectivos, podendo servir de inspiração a alguns aspectos do protótipo. Em resumo, as vantagens da abordagem deste trabalho poderão ser introduzidas directamente nos conceitos ACM, sem ser necessário a adopção deste modelo menos convencional, e a lógica de comparação entre livros, cursos, disciplinas, é algo que poderá ser adoptado como resultado do modelo computável.

Por fim, os trabalhos apresentados sobre listas referenciais de competências (Dublin Descriptors, Dublin Descriptors Eng. e competências definidas no D.I.) são de especial importância para o modelo e a sua forma de recolha / representação desta informação. Uma competência, qualificação ou *skill* a fornecer ou promover junto do aluno, já foi identificado como uma peça fundamental ao nível da definição curricular, fazendo por isso sentido ter várias formas de a referir e eventualmente classificar. Por exemplo, os LOs do BoK ACM são uma forma bastante detalhada de identificar as competências que a frequência de determinada disciplina (que aborda certos tópicos) transmite ao aluno, mas no entanto estas poderão não ser as únicas competências que a frequência da disciplina promove (por exemplo, *soft-skills*). Adicionalmente, além da análise de DDs e DDs-e, a análise do trabalho de adequação do curso do D.I. a Bolonha, introduz também formas de relacionar estas competências. Isto fornece pistas para uma eventual base de relações de competências, que permita ao modelo computável a disponibilização de visões de cobertura das competências a vários níveis (detalhe em LOs, mais abrangente em DDs ou DDs-e, etc.). A taxonomia de *Bloom* é algo que poderá ser utilizado na classificação e distinção dos verbos utilizados nas frases identificativas das competências, em particular sobre os objectivos de aprendizagem ACM.

Capítulo 3
Apresentação e Detalhe do Modelo

Uma descrição detalhada sobre o modelo proposto nesta dissertação, onde se efectua a descrição do seu âmbito e objectivos, se apresenta um visão geral introdutória, seguida de uma análise detalhada das várias componentes da sua estrutura. Descrevem-se aspectos relativos à sua extensibilidade, bem como do seu povoamento com informação. Completa-se com uma descrição de possíveis explorações do mesmo.

Tendo por base os conceitos dos trabalhos descritos no capítulo 2, é possível abordar os processos de construção, objectivos, características e eventuais aplicações do modelo proposto nesta dissertação. Pretende-se que este seja uma possível abordagem a um modelo computável, de acordo com as características referidas no capítulo 1, para apoio ao problema da definição curricular dentro da área da ciência da computação.

Relativamente à estrutura deste capítulo, na secção 3.1 descreve-se o âmbito e objectivos do modelo, fazendo nas secções 3.2 e 3.3, respectivamente, uma apresentação geral mais introdutória à sua estrutura, e outra mais detalhada, às suas componentes principais. Na secção 3.4 serão abordados aspectos acerca do povoamento de informação no modelo, enquanto que por fim, na secção 3.5, são descritas várias formas de exploração do modelo proposto.

3.1 Âmbito e objectivos do modelo

Tendo presente o descrito no capítulo 1.3 (onde se aborda o âmbito e objectivos da tese), identificam-se nesta secção o âmbito e objectivos específicos do modelo.

Pela adopção de um corpo de conhecimento (*body of knowledge*) baseado em standards, bem definido e com capacidade de relacionamento entre os seus conceitos, pretende-se:

1. O estabelecimento de um formato para a estruturação das temáticas de conhecimento na área de CS;
2. A garantia de uma normalização do corpo de conhecimento de forma a que seja possível a sua eventual partilha com outros estabelecimentos de ensino;
3. A capacidade de extensão e evolução dessa estrutura internamente ao DI, bem como a sua "alimentação" com base em eventual informação externa.

Pela definição de conceitos que mapeiam as conhecidas noções de disciplina e versão de disciplina[20], curso, estrutura, plano e oferta curricular, tendo por base a actual Licenciatura em Eng.ª Informática (1º ciclo) no DI, pretende-se:

4. O estabelecimento de um formato para recolha da informação estrutural de disciplinas e para recolha da informação estrutural de um curso.

[20] Entenda-se aqui versão, como uma instanciação de disciplina, baseada na sua definição. Por exemplo, Programação II leccionada no ano 2009-2010, que poderá ser ligeiramente diferente da leccionada no ano anterior, mas que seguirá o mesmo "padrão" definido.

5. A garantia que estes formatos são compatíveis (ou aproximadamente compatíveis) com o que já existe no DI ao nível de definição curricular.
6. Que o modelo proposto, por ter uma base num exemplo concreto real, tenha uma eventual aplicabilidade concreta e efectiva à definição curricular.

Pela existência do corpo de conhecimento (e seus objectivos) e da sua utilização como base da definição do conceito que mapeia disciplina, pretende-se:

7. Uma definição compatível e normalizada de todas as disciplinas, garantindo a capacidade de comparação e cruzamento entre as informações estruturais das mesmas.
8. Uma capacidade de partilha e comparação da informação curricular com outros estabelecimentos de ensino.

Pela adopção no modelo de uma separação entre os aspectos de estrutura e operacionais (a dualidade entre especificação de disciplina e versão de disciplina), pretende-se:

9. Promover a importância da definição dos aspectos estruturais de um curso e respectivas disciplinas: a completude e cobertura dos temas e competências a atingir.
10. Minimizar o impacto dos aspectos da operacionalização anual dos cursos e das diferentes versões das disciplinas, garantido a evolução normal do ensino e a manutenção de histórico.
11. Permitir que os diferentes actores envolvidos na definição curricular do curso e das disciplinas, se foquem nos principais aspectos que são da sua efectiva responsabilidade.

Pela definição clara e aberta do modelo proposto, numa abordagem e perspectiva modular, pretende-se:

12. Permitir que o modelo seja "relacionável" ou faça uso de outras fontes de informação. Por exemplo, outros sistemas dentro da Universidade, que sejam também repositórios de informação relevante para a definição curricular.
13. Inversamente, permitir que a informação existente no modelo seja também partilhável para outros modelos ou sistemas. Ou seja, a possibilidade de serem outros sistemas a adoptar e consumir a informação do modelo.

Por fim, pelo simples facto da existência do modelo proposto para recolha de informação das diversas componentes envolvidas na definição curricular, pretende-se:

14. Abordar um dos principais focos da dissertação, garantindo que pelo facto da informação ser recolhida por um modelo computável, com uma estrutura conhecida e bem definida, consegue-se usufruir de todas as vantagens de ter informação computável no apoio à definição curricular. Considere-se por exemplo, as vantagens enunciadas na secção 1.1.3 (a motivação), de onde se pode referir a identificação de perfis dos alunos com base nas competências obtidas, a eventual comparabilidade (ao nível temático) entre planos e disciplinas, etc.

3.2 Apresentação geral do Modelo

O modelo que se apresenta neste capítulo é constituído por diversas componentes, que deverão "focar-se" em distintas partes do problema da definição curricular. Esta secção apresenta o modelo numa visão mais geral, alto nível, referindo-se primariamente às suas grandes áreas, antes de serem abordados os aspectos de detalhe nas secções posteriores.

3.2.1 Principais conceitos e nomenclatura

Antes da apresentação geral do modelo, é importante introduzir e clarificar alguns conceitos que serão utilizados ao longo do documento e que referem aspectos centrais do modelo e da solução.

Chama-se à atenção para a nomenclatura adoptada neste capítulo, que fará uso de expressões em português para referir os vários conceitos do modelo, sendo no entanto o inglês utilizado quando se pretende indicar, por exemplo, algum conceito reutilizado dos trabalhos ACM. Porém, refere-se que na implementação concreta deste modelo, foi considerada uma adopção total do inglês, antevendo eventuais necessidades de partilha do modelo.

Corpo de conhecimento (CdC), é uma expressão já abordada no documento que, quando utilizada no modelo e na solução proposta, refere um conjunto bem definido de conhecimento dentro da área, disponível para ser abordado num curso e respectivas disciplinas. Este corpo de conhecimento, que é constituído por diversas divisões e relações entre si, permite: *i*) A indicação (via referência) de quais os temas, tópicos, etc., que são abordados ou estão relacionados com os vários constituintes que fazem parte da definição de disciplinas e cursos; *ii*) A indicação de vários tipos de *skills* envolvidas nos processos de ensino relacionados com as disciplinas; *iii*) Ter uma base de informação acerca de currículos ou definições curriculares mínimas. Definiu-se este corpo de

conhecimento baseado nos trabalhos apresentados anteriormente, em concreto o CC2001 [28] e subsequentes trabalhos com sugestões de melhorias ao mesmo. Considerando esta base nos trabalhos ACM e relacionados, e conforme referido acima, serão utilizadas algumas expressões em inglês dentro do corpo de conhecimento que deverão ser interpretados como conceitos reutilizados desses trabalhos.

O que até aqui tem sido mencionado como disciplina ou *course* (em inglês) [29], refere-se à unidade ou componente mais simples do *curriculum* de um curso. Esta nomenclatura de **Unidade Curricular (UC)** é definida no processo de Bolonha como uma "Unidade de ensino com objectivos de formação próprios" [20], e será também adoptada ao longo deste documento para se referir à representação deste conceito no modelo. É também considerada a versão em inglês, *Curricular Unir*, que será também utilizada quando o modelo é totalmente utilizado em inglês, por exemplo, na sua implementação no protótipo.

Aplicando a mesma lógica anterior à definição curricular, é necessário ter presente que um curso não é unicamente uma lista organizada de unidades curriculares. Além da organização das unidades curriculares em anos e semestres, será necessário considerar também os aspectos relativamente às unidades curriculares obrigatórias e opcionais, áreas científicas com créditos mínimos, etc. Assim sendo, seguindo a nomenclatura utilizada no ensino, em particular no Departamento de Informática [87], foram considerados 3 principais conceitos: *i*) **Estrutura Curricular (EC)** como a organização conceptual de um curso, onde se definem quais as unidades curriculares obrigatórias, quais os grupos de unidades curricular opcionais, quais os créditos mínimos, etc.; *ii*) Directamente relacionado com a EC, o **Plano Curricular (PC)**, ou também conhecido como "Plano de Estudos" [88], será a distribuição das unidades curriculares (já associadas e definidas na estrutura curricular) pelos vários anos e semestres de um curso, tipicamente; *iii*) Por fim, a noção de **Oferta Curricular (OC)** corresponde a aplicação do plano curricular nas suas várias versões anuais, ou seja por exemplo, a oferta curricular do ano 2009-2010, onde algumas das ofertas do plano curricular poderão não existir. Assim, utilizando como base estes 3 principais conceitos, modelam-se também outras estruturas conceptuais, como a noção de curso, perfil, ramo, etc.

Tendo por base estes conceitos, é possível então apresentar uma visão mais da estrutura do modelo proposto. Refere-se também o anexo B (digital), que contém detalhe sobre os processo propostos para o modelo.

3.2.2 Visão geral do modelo

Como introdução é importante ter presente que o modelo pretende aplicar-se à área CS, podendo no entanto ser definido com alguns conceitos eventualmente mais genéricos e possivelmente extensíveis ou aplicáveis a outras áreas. Como forma de promover esta característica de extensibilidade e a real computabilidade

do modelo, este deverá ter presente o objectivo da normalização de informação sempre que possível.

Assim, considerado que o modelo existirá para apoiar a definição curricular, faz sentido que cubra os processos referidos na secção 1.3 (apresentados em particular na Figura 1.4), originando assim 3 grande áreas que são consistentes com os conceitos apresentados atrás:

i. **(Definição da) Temática e objectivos de aprendizagem**
ii. **(Definição das) Unidades curriculares e suas versões**
iii. **(Definição da) Estruturação curricular e suas versões**

Estas 3 grandes áreas poderão ser representadas conceptualmente na Figura 3.1, esboçando-se já alguns dos assuntos que estão envolvidos em cada uma.

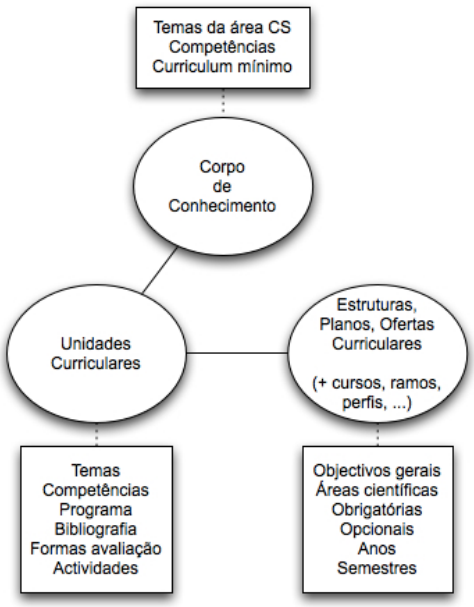


Figura 3.1 – As grandes áreas do modelo

É importante ter presente que quando se referem as unidades curriculares (UC), menciona-se também "e as suas versões". Isto deve-se ao facto deste conceito ter geralmente uma dualidade associada, um aspecto de definição conceptual e um aspecto mais operacional relacionado com a sua execução. Para suportar esta necessidade, considerou-se a introdução de 2 conceitos:

- **Modelos** (as Unidades Curriculares e as Estruturas Curriculares)
  É importante não confundir este conceito com o modelo que se está a descrever, dado que esta nomenclatura "modelo" aplica-se a um conceito do modelo computável que se está a propor. A ideia é que seja uma UC ou EC, estes são efectivamente "modelos", onde se definem objectivos e características alto nível ou conceptuais. Por exemplo, um modelo de UC (ou referido unicamente como UC), definirá uma espécie

de "template" com as componentes conceptuais mais relevantes já detalhadas para aquela disciplina em particular (por exemplo, temas e objectivos de aprendizagem). Este servirá de base para a utilização ao longo da evolução da disciplina, seja em vários cursos e vários anos, e poderá ser considerado como a "versão mestre" para ser utilizada a quando da definição das suas efectivas versões. Naturalmente estes modelos ou "versões mestres" deverão considerar a possibilidade de evoluir também.

- **Instâncias** (de Unidades Curriculares e as Ofertas Curriculares)
  As instâncias podem ser resumidas como as efectivas versões operacionais dos modelos. Isto é, se os modelos são "templates", as instâncias são as corporizações dos templates, com introdução dos detalhes mais operacionais, não relativos aos objectivos e organização alto nível. Por exemplo, imagine-se que para uma disciplina de "Introdução à Programação" é definida uma UC onde se declararam quais os tópicos a abordar (entre outras coisas). Para uma versão dessa disciplina no ano 2009-2010, deverá ser definida uma instância baseada na UC, que utilizará essa base definida, acrescentando detalhes relativos à execução "prática" do ensino da disciplina (bibliografia, etc.). Uma lógica similar, mas não necessariamente igual, é aplicada às estruturas curriculares e às ofertas curriculares: enquanto as primeiras são modelos relativos à definição de um curso, as segundas correspondem ao conceito das ofertas de unidades curriculares (instâncias) disponíveis várias instanciações anuais dos cursos. Apesar da relação entre modelos e instâncias, estes serão definidos elementos distintos, que se focam em diferentes aspectos do ensino, a definição conceptual e a execução ou operacionalização.

Representa-se assim na Figura 3.2 uma visão geral mais detalhada do modelo, com introdução das distinções entre "modelo" / "instância" e os respectivos tipos de informação que ambos permitirão definir, assim como um detalhe dos constituintes do corpo de conhecimento proposto.

Conforme se pode verificar de imediato, as 3 grandes áreas da Figura 3.1 (corpo de conhecimento, unidades curriculares e estruturas / planos / ofertas curriculares), estão representadas aqui como secções com um padrão de fundo cinzento claro, passando a ser grupos ou divisões abstractas que englobam um conjunto de conceitos.

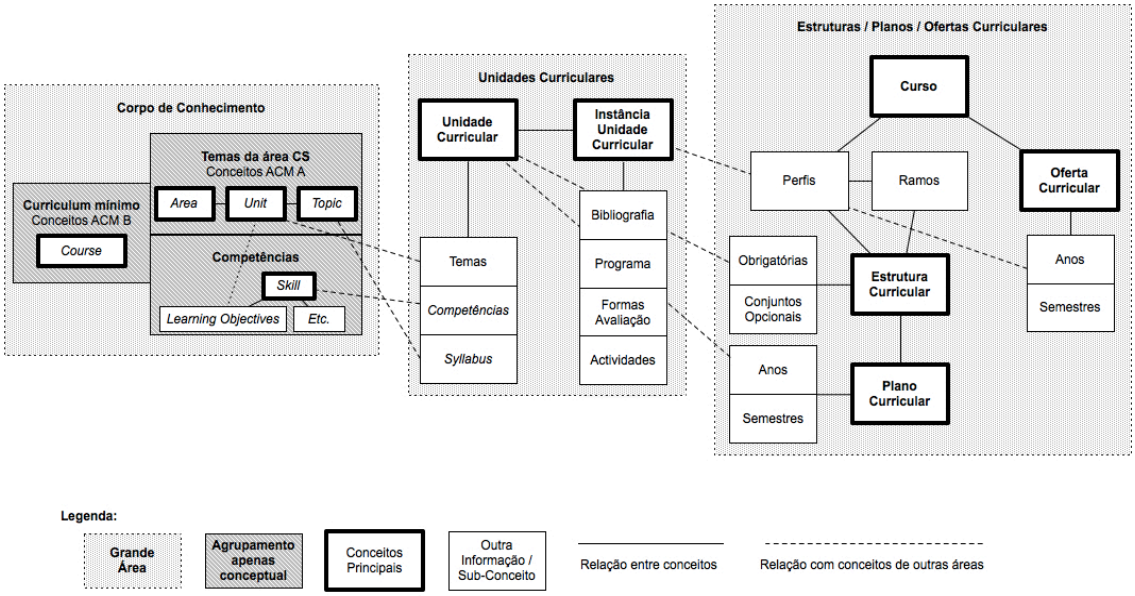


Figura 3.2 – Visão conceptual alto nível do modelo proposto

Seguindo a legenda, dentro da secção "Corpo de Conhecimento" estão representados mais 3 agrupamentos conceptuais, que sendo grupos abstractos de conceitos que têm algo em comum, são utilizados apenas nesta figura para facilitar a compreensão do modelo e indicar que existe uma divisão lógica dentro do mesmo. Por exemplo, o agrupamento das "Competências" dentro do corpo de conhecimento, contém vários conceitos, sendo *Skill* o principal. Na sequência da legenda, os rectângulos com a borda a negrito representam os conceitos mais relevantes do modelo, tipicamente mapeados como peças concretas no modelo e na sua implementação. Os rectângulos normais (i.e., não a negrito), representam subconcedidos que retratam outros tipos de informação, geralmente associada com os conceitos principais. Quando existe algum tipo de relação entre conceitos que se considerou relevante para esta visão geral, representa-se a mesma com um traço simples, o que permite dar uma imagem inicial das relações no modelo. Por fim, as ligações representadas a tracejado indicam relações mais relevantes entre conceitos das grandes áreas.

Cada conceito principal apresentado pode ser resumido seguinte forma:

- **Corpo de conhecimento**
  Definido com base na utilização dos conceitos apresentados nos vários trabalhos referidos no Capítulo 2, assumiram-se as 3 divisões conceptuais referidas atrás e representadas na figura: *i*) uma lista extensa (mas não completa) de *knowledge areas* da CS, que por serem maioritariamente baseadas no anexo A do CC2001, se passa a designar por *BoK* ACM-A (ou unicamente ACM-A); *ii*) uma lista de definições base de UC para um currículo mínimo da área CS, expressas conforme o anexo B do CC2001 (portanto *courses*), que se passa a designar por *BoK* ACM-B (ou unicamente ACM-B); *iii*) um conjunto de várias listas de

competências, baseadas nos trabalhos da conversão de Bolonha no D.I., no CC2001, nos *Dublin Descriptors*, etc., englobadas no conceito genérico *skill* e que passa a designar por *BoK* SKILL (ou unicamente SKILL). Note-se a utilização da nomenclatura em inglês, pelo facto de se estar a referir conceitos dos trabalhos ACM. Descrevem-se cada uma das divisões:

- **ACM-A**
  Utilizando a nomenclatura e dados definidos pelo CC2001, foi estabelecida uma hierarquia *areas* → *units* → *topics*, com um conjunto de relações entre si. Em cada *unit* estão também associados os *learning objectives* respectivos, que serão incluídos como parte de SKILL.
- **ACM-B**
  Utilizando a nomenclatura e dados definidos pelo apêndice B do CC2001, estabeleceu-se uma base de informação curricular mínima com a reutilização do conceito *course*, que é definido com recurso à informação ACM-A. Este conceito poderá ser encarado como um base de apoio à definição de UC.
- **SKILL**
  Tendo presente o conceito *learning objectives*, originário do CC2001, e um conjunto adicional de formas de definição de competências (*Dublin Descriptors,* etc.), estabeleceu-se o conceito global *skill,* que permite identificar qualquer competência no modelo. Tendo presente necessidades de evolução, este conceito pode conter relações entre si e ser classificado de acordo com o seu tipo, existindo assim *skills* do tipo *learningObjectives*, *dublinDescriptors*, etc.

- **Unidades Curriculares**
  Cada conceito UC representa um "template" de cada uma das UC disponíveis no curso de informática. Como "template", entende-se uma definição organizativa orientada à estruturação temática e conceptual da UC, sendo por isso constituída: *i*) por **temas** de conhecimento a abordar (baseados nas *units* e *topics* ACM do *BoK* ACM-A); *ii*) por **competências** que se pretende que o aluno obtenha após frequentar a UC (baseadas nas *skills* do *BoK* Skill); *iii*) e por um ***syllabus*** que representará a organização (ordem, detalhe, etc.) dos temas a abordar na UC (baseado nos *topics* ACM do *BoK* ACM-A).

- **Instâncias de Unidades Curriculares**
  Directamente relacionados com os modelos, as instâncias de UC corporizam questões mais operacionais da UC, sendo constituídas: *i*) por uma **bibliografia** recomendada; *ii*) um **programa** contendo uma

organização estruturada e ordenada dos tópicos a abordar (tendo por base o descrito no *Syllabus* do conceito UC); *iii*) uma descrição das principais **actividades**, além de aulas típicas, que terão alguma importância na aprendizagem dos temas ou contenham actividades de avaliação; *iv*) e uma descrição de alto nível do modelo ou **formas de avaliação** presentes nesta versão / instância da UC

- **Estrutura Curricular e Plano Curricular**
  Uma Estrutura Curricular (EC), de forma similar à UC, representa um *template* que permite definir os aspectos estruturais de um *curriculum*. Em concreto, pode-se definir o ***curriculum* mínimo obrigatório** e os **conjuntos/alternativas de disciplinas opcionais**, tendo por base as UC (modelos) já definidas. Adicionalmente, pode ser definido um Plano Curricular (PC) associado, que utilizará as UC indicadas na EC para estabelecer a organização recomendável desse *curriculum*, com a **distribuição anual e semestral** sugerida para as mesmas. Estes conceitos de EC e PC, na verdadeira acepção da palavra *template*, não são por si só elementos no modelo, mas na realidade bases para a criação de outros conceitos que permitem estruturar um curso, como os perfis e ramos, referidos em pontos mais à frente.

- **Oferta Curricular**
  Considerando que as EC e os PC permitem definir os aspectos estruturais do *curriculum*, utilizando as UC para esse efeito, o conceito Oferta Curricular (OC), utilizando as instâncias de UC, permite a instanciação desses aspectos estruturais nas várias versões anuais do curso. Em concreto, cada OC corresponde à distribuição, seguindo a habitual estrutura anual e semestral, de todas as instâncias de UC disponíveis nessa período do curso. Por exemplo, a oferta curricular do ano 2009-2010, consiste num plano curricular instanciado com as UC (instâncias) que estarão disponíveis aos alunos nesse ano lectivo.

- **Curso – Versão, Perfil e Ramo**
  Conforme pode ser verificado na Figura 3.2, está representado o conceito **curso**, que naturalmente se associa à noção que geralmente temos de um curso, tendo associado um nome, descrição, tipo (ex: 1º, 2º ciclo) e objectivos a atingir. De forma a lidar com as diferentes formas de especificação de *curriculum* existentes nas faculdades (e em particular no DI), considerou-se não apenas um conceito linear, mas a noção que um curso poderá ter diferentes versões, que por si poderão ter um ou mais perfis, e subsequentemente, estes poderão ter ou não ter vários ramos. O conceito de **versão** de um curso, apesar do nome, não está directamente relacionada com as várias iterações anuais, mas

sim com diferentes iterações de especificação curricular que um curso poderá ter sofrido ou vir a ter. Considere-se por exemplo, a reestruturação que o curso sofreu da versão Pré-bolonha para a versão Bolonha, sendo a mesma licenciatura, foi reestruturado de 5 anos para 3 anos, sofrendo naturalmente afectações ao nível das UC oferecidas, nos objectivos, nas competências a atingir, etc. Em cada versão do curso, considera-se que poderá existir um ou mais distintos **perfis** curriculares, sendo estes uma extensão do conceito de Estrutura Curricular. Portanto, sendo estes também EC, permitem definir diferentes *curriculum* para diferentes perfis dentro do mesmo curso. Como exemplo, veja-se o exemplo o DI, onde existe o perfil "Ciências de Engenharia" e "Informática Aplicada". De forma a lidar com aplicações do modelo a outros cursos que não apenas no DI, ou eventualmente futuras evoluções do curso, considerou-se que poderão existir **ramos** dentro de cada perfil. Um ramo é também uma extensão do conceito Estrutura Curricular, o que permitirá a definição de diferentes *curriculum* (ou alterações) ao estabelecido no perfil. Veja-se a Figura 3.3, onde se apresenta o que a hierarquia e relações descritas acima, em maior detalhe do que o apresentado na Figura 3.2.

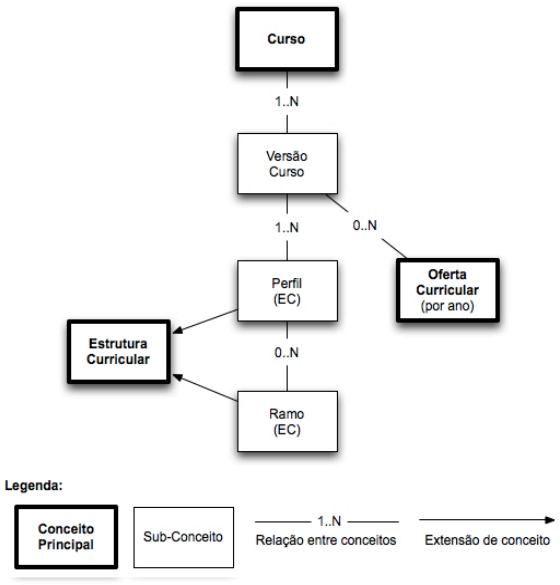


Figura 3.3 - Detalhe da visão geral do conceito Curso

Como já foi referido, esta visão geral de alto nível apresentada não detalha todos os aspectos, atributos e relações existentes, sendo efectuada uma explicação em maior detalhe de cada elemento na secção 3.3. Refere-se o anexo B (digital), que contém algum detalhe adicionais sobre os processos e actores envolvidos no modelo.

3.3 Apresentação detalhada da estrutura principal do Modelo

Nesta secção, conforme o título, é apresentada uma descrição mais detalhada do modelo proposto. Para cada conceito alto nível já abordado na visão geral, existirá aqui uma subsecção respectiva, onde serão abordados aspectos de detalhe da construção do modelo, em concreto, quais os seus sub-conceitos, atributos e relacionamentos respectivos. Consideram-se então as 5 seguintes secções: 1) CdC – *ACM-Appendix* A; 2) CdC – *ACM-Appendix* B; 3) CdC – *Skill*; 4) Unidade Curricular; 5) e por fim, Curso e Estrutura / Plano / Oferta Curricular.

Sabendo que alguns dos conceitos, em particular os do CdC, são baseados noutros trabalhos já anteriormente descritos, considerou-se importante criar uma distinção entre estes e os conceitos definidos de raiz na dissertação. O objectivo é a identificação de "zonas" (ou seja, conjuntos de conceitos), que poderão ser futuramente alimentadas com nova informação, ou eventualmente partilhadas para outros modelos, sistemas, etc. Assim, paralelamente às grandes áreas ou zonas do modelo, optou-se por dividir conceptualmente o modelo nas 3 divisões abstractas do CdC (referidas na secção 3.2) e num grupo de conceitos internos: os grupos *acmA*, *acmB*, *skills* e o grupo de conceitos definido para o DI. Com objectivo de identificar o grupo a que cada conceito pertence, propõe-se utilizar uma nomenclatura para referir e identificar os conceitos, inspirada na utilização de *namespaces* XML [92]:

Grupo : Nome_Do_Conceito_Ou_Atributo_Ou_Relação

Note-se que, inversamente ao apresentado na visão geral, optou-se por utilizar o totalmente o inglês na modelação dos vários conceitos, pois permite uma maior facilidade numa eventual ligação com outros repositórios de conhecimento (que poderão não ser nacionais). Como exemplo, "*acmA:area*" refere um conceito que representa uma área temática dentro do CdC, mas que pertence ao grupo de conceitos do apêndice A do ACM. Ou seja, é um conceito existente dentro do CdC, mas identificado no grupo dos conceitos ACMA. Outro exemplo, "*di:curricularUnit*", refere o conceito de UC, definido dentro grupo "DI", que agrupará os vários conceitos definidos especificamente no âmbito desta dissertação.

Adicionalmente, como forma de representar os detalhes do modelo, optou-se por seguir um formato mais ou menos intuitivo, aproximadamente equivalente à representação de modelos de informação semânticos [93]. Assim, apresentam-se na Figura 3.4 os principais elementos utilizados para a representação do modelo, explicitando as diferenças entre conceito, atributo e relação.

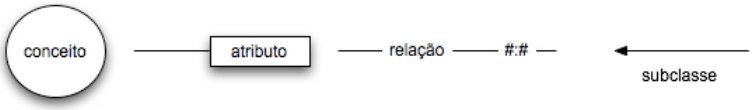


Figura 3.4 – Legenda da representação do modelo proposto

Detalhadamente, um conceito, representado por um círculo, identifica o elemento de informação que deverá existir no modelo, baseado na noção descrita. Estes elementos poderão conter atributos de informação com dados concretos ou relações para outros conceitos. Um atributo é representado por um rectângulo. Uma relação, que é representada por uma "linha" ou ligação entre conceitos, indica que existe uma relação de determinado tipo entre 2 ou mais conceitos. As relações, além da identificação pelo nome (que existirá sempre), poderão conter informação de cardinalidade entre os conceitos, sendo aplicável o mesmo tipo de conceito de cardinalidade do modelo "entidade-relação" [94] ou dos modelos UML [90]. Da mesma forma, assume-se que quando não é declarada cardinalidade, a relação será de 1 para 1. A relação deverá ser lida sempre do lado do nome da relação para o lado da cardinalidade, sendo representada sempre nas figuras do conceito que a detêm e que se liga ao outro conceito (ou seja, uma relação na figura do conceito A com ligação a B, é uma relação de A para B). Por fim, existirá ainda a possibilidade da aplicação da noção de herança [95] aos conceitos, ou seja, a subclasse ou extensão de um conceito por outro conceito, que será representada por uma seta da subclasse para a classe base.

É importante referir que não serão descritos todos os pormenores e detalhes do modelo, mas sim um *overview* detalhado dos conceitos daquela secção. O detalhe completo do modelo, ou seja, a descrição muito completa dos atributos, conceitos e relações, estarão resumidos num documento anexo digital a esta dissertação, com a referência de anexo B - Detalhe do Modelo. Ao longo das secções seguintes serão feitas referências para este documento, indicando onde se poderão verificar mais detalhes acerca do conceito abordado na secção.

3.3.1 CdC - ACM-Appendix A

Descreve-se aqui a proposta da componente ACM-A do CdC, que representará a informação dos temas CS disponíveis aos vários outros conceitos do modelo.

O modelo para a componente ACM-A, conforme já referido anteriormente, propõe a utilização da hierarquia do trabalho ACM CC2001 [29], que pode ser resumida a:

AREA → UNIT → (TOPIC | LearningObjectives)

Após o estudo de várias hipóteses de definição do modelo, optou-se por utilizar também os conceitos propostos nos trabalhos da Universidade de Milão,

em concreto o trabalho "*Hyperkrep academic communities*" [61], introduzido também a noção base de ITEM e de várias outras relações. Pode-se verificar na Figura 3.5 uma representação desta componente e seus vários conceitos.

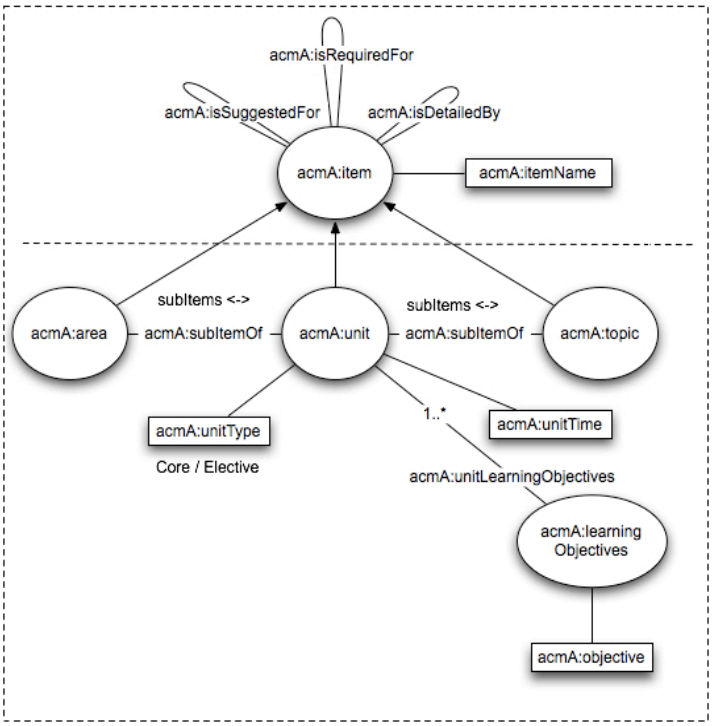


Figura 3.5 - Detalhe do modelo da componente CdC - *ACM Appendix* A

Os principais conceitos do modelo italiano estão representados na parte superior da figura, enquanto que na parte inferior estão directamente mapeados os conceitos ACM CC2001. A escolha do modelo italiano deve-se essencialmente a 3 pontos:

i. A simplicidade com que evolui o modelo CC2001, sem causar grandes alterações ao mesmo (e portanto mantendo-se estável para futuras integrações de informação);

ii. A capacidade futura de extensão / evolução dos conceitos, devido à utilização de um elemento genérico ITEM;

iii. A inclusão de relações adicionais, não originalmente propostas no CC2001, que permitem estabelecer outras co-relações entre *areas*, *units* e *topics*, facilitando a sua pesquisa, cruzamento, etc., assim introduzindo mais informação e possibilidades de cruzamento no CdC.

O elemento ***acmA:item*** representa um conceito abstracto (similar à utilização de uma classe abstracta na programação por objectos [95]) que servirá de base para todos outros conceitos ACM. Serão aqui estabelecidas as relações e atributos, que serão utilizados nos objectos conceptuais ACM (*area*, *unit*, *topic*), como por exemplo o atributo *acmA:itemName*, que conforme o nome indica, permite identificar o conceito com um nome inequívoco. As diferentes relações existentes, aplicáveis aos conceitos ACM, permitem indicar vários tipos ligações

entre os diferentes conceitos, estando a sua semântica directamente associada ao seu nome:

- A relação *acmA:isSuggestedFor* refere-se a relações do tipo "é também sugerido conhecimento de" relativamente à aprendizagem dos conceitos ACM *areas*, *units* ou *topics*.
- A relação *acmA:isRequiredFor* refere-se a relações de "é requerido saber sobre" relativamente à aprendizagem dos conceitos ACM. Note-se que serve apenas para referência e não implica necessariamente a obrigatoriedade deste requerimento na definição de curricular.
- A relação *acmA:isDetailedBy* refere-se a relações do tipo "é detalhado em maior pormenor por".

O elemento ***acmA:area*** representa o conceito que é o topo da hierarquia do CdC *ACM* e que se entende por área de conhecimento dentro da ciência de computação. Como exemplo, aliás já referido anteriormente, a área de estudo de "algoritmos e complexidade", é representada no CdC por um item de nome "AL. *Algorithms and Complexity*" que engloba um conjunto de informação (*units* e *topics*) sobre esse mesmo tema.

Note-se que as duas relações apresentadas na figura entre *area* e *unit,* e entre *unit* e *topic*, tem igualmente relações inversas:

- A relação *acmA:subItemOf* expressamente indicada na figura, na direcção *unit* → *area* e *topic* → *unit*, que indica que determinada *unit* é "filha" de *area,* ou que determinado *topic* é filho de *unit*;
- E a relação *acmA:subItems*, inversa à anterior, na direcção *area* → *unit* e *unit* → *topic*, que indica quais os elementos filhos directos de *area* (ou seja, quais as *units*) e de *unit* (ou seja, quais os *topics*).

Na continuação, o elemento ***acmA:unit*** representa naturalmente o conceito *UNIT* ou unidade temática, que é o elemento do meio da hierarquia CdC ACM, existindo como "filhos" directos das *areas*. Seguindo a proposta ACM CC2001, estes representam conceptualmente "módulos temáticos individuais" [29] dentro das *areas*, sendo também a forma principal para indicação de temas a abordar nas Unidades Curriculares (UC). Existirá um atributo *acmA:unitTime,* indicando o tempo mínimo estimado em horas para a aprendizagem da unit, um atributo *acmA:unitType* que indica se a *unit* faz ou não parte *core* do curriculum mínimo obrigatório de temas para um estudante CS, e uma outra relação além das indicadas acima, de nome *acmA:unitLearningObjectives*, que permite ligar a *unit* aos elementos que representam os objectivos de aprendizagem a si associados.

No final da hierarquia temos o conceito *TOPIC*, representado pelo elemento ***acmA:topic***, o "grão mais fino" do conhecimento CS.

Por fim, existe também o conceito ***acmA:learningObjectives*** que representa o conceito objectivos de aprendizagem. Segundo a proposta ACM, estando estes

ligados a uma *unit* em particular, podem ser entendidos como capacidades, competências, *skills*, que um aluno adquire após frequentar essa *unit*. Para sua identificação, existe o atributo *acmA:objective*, onde se descreve o objectivo de aprendizagem na forma textual. Por fim, pelo facto destes objectivos referirem *skills* de diversos tipos (conhecimento, prática ou *soft-skills*), de imediato assume que deverão efectivamente fazer parte do grupo de elementos referido anteriormente como "*Skills"*. No entanto, por uma questão de coerência com o formato ACM, optou-se por considerar o conceito *acmA:learningObjectives* como fazendo primeiramente parte do "ACM-A" (mantendo a nomenclatura), mantendo ligações por parte dos conceitos do grupo "Skills". Conceptualmente pode-se encarar o conceito como existente nos 2 grupos, conforme a Figura 3.6, apesar de ser modelado no primeiro grupo.

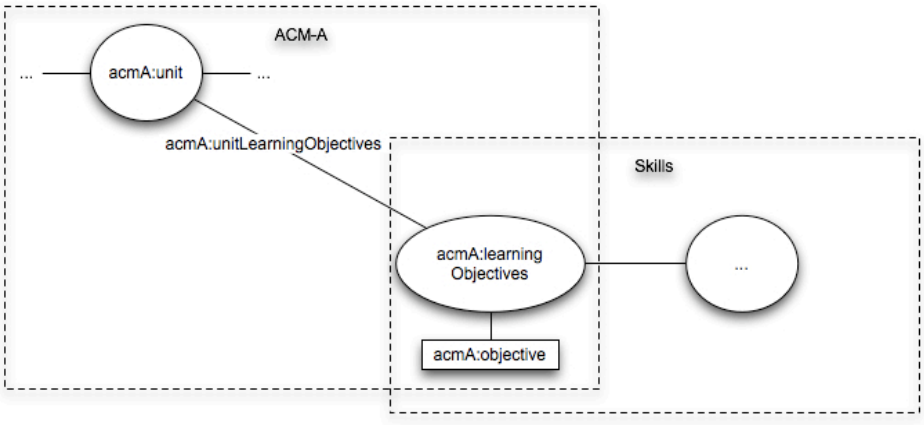


Figura 3.6 – O conceito *acmA:learningObjectives* no grupo "ACM-A" e sua relação com o grupo "*Skills*"

3.3.2 CdC - ACM-Appendix B

Descreve-se aqui a proposta da componente ACM-B do CdC, apresentada na Figura 3.7, que representará o modelo proposto pelo ACM para a definição de *courses,* já anteriormente abordado na secção 2.2.3.2 (relativa ao apêndice B do CC2001).

A utilização directa do modelo ACM (com apenas alguns ajustes de detalhe), "alimentado" posteriormente com o conjunto de disciplinas (*courses* conforme a nomenclatura CC2001) já definidas e detalhadas no apêndice B do CC2001, permite criar uma base ou corpo de conhecimento sobre o currículo mínimo recomendado pelo ACM. Este currículo mínimo, por ter sido definido e aprovado pelo ACM, é naturalmente uma boa base para quer para o apoio à definição de unidades curriculares (focando-se apenas num *course*), quer para o apoio à definição de outras estruturas curriculares, permitindo eventualmente a comparações entre si.

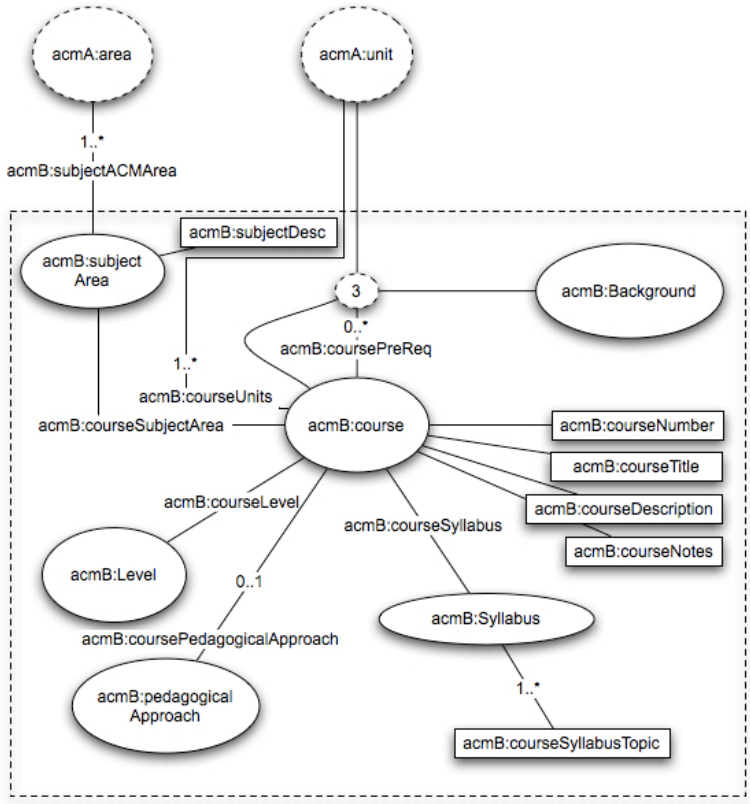


Figura 3.7 - Detalhe do modelo da componente CdC - *ACM Appendix B*

O modelo centra-se no conceito ***acmB:course***, que corresponde à proposta ACM para a definição daquilo que tipicamente se considera como uma disciplina num curso (ou unidade curricular, dada o ênfase na definição e não operacionalização). Contém alguns atributos que permitem identificar e definir univocamente cada uma das disciplinas: *i*) *acmB:courseNumber*, uma nomenclatura alfanumérica para proposta pelo ACM para identificação de *course* e anteriormente referida na Figura 2.15 da secção 2.2.3.2; *ii*) *acmB:courseTitle* e *acmB:courseDescription*, respectivamente o nome/título e descrição textual do âmbito e objectivos da disciplina; *iii*) e por fim, *acmB:courseNotes*, para notas adicionais e esclarecimentos. Existem relações de *course* para outros conceitos, que podem ser divididas em 2 tipos: relações de classificação e relações relativas ao conteúdo temático.

As relações de classificação refere-se a aspectos de organização da disciplina dentro do currículo exemplo mínimo ACM e estão directamente relacionadas com detalhes da nomenclatura do atributo *acmB:courseNumber*. A relação *acmB:courseLevel*, que liga *course* ao elemento ***acmB:Level***, permite indicar qual o nível da disciplina dentro de um eventual currículo mínimo, isto é, se é uma disciplina introdutória, intermédia, avançada ou de projecto. A relação *acmB:courseSubjectArea*, que liga a ***acmB:SubjectArea***, permite classificar a disciplina de acordo com os temas que são abordados, seguindo um agrupamento proposto pelo ACM (baseado de forma mais ou menos linear nas *areas* ACM). O CC2001 apenas aplica esta classificação a disciplinas de nível intermédio e

avançando, pelo que se manterá esse comportamento para compatibilidade, não se prevendo a utilização desta relação para classificar futuras unidades curriculares a definir. O *acmB:subjectArea* introduz uma relação (*acmB:subjectACMArea*) para as *areas* referidas em *acmB:course* e uma descrição textual da *subject area* em questão (*acmB:subjectDesc*). Por fim, existe ainda também outra relação de classificação referida como *acmB:coursePedagogicalApproach*, que liga *course* a ***acmB:PedagogicalApproach*** e que permite indicar a estratégia ou modelo de organização da disciplina dentro do plano curricular, ou seja, conforme anteriormente abordado na secção 2.2.3.2, qual a estratégia de implementação curricular proposto pelo ACM onde a disciplina se enquadra ("*Imperative-first*", "*Object-First*", etc.). Novamente, será uma relação que existe por compatibilidade com o modelo ACM não se prevendo directamente a sua utilização na definição de unidades curriculares adicionais.

Naturalmente, as 3 restantes relações permitem definir o conteúdo temático e âmbito da disciplina, estabelecendo relações, conforme se pode verificar na Figura 3.7, com o corpo de conhecimento ACM-A e outros elementos específicos. A primeira e eventualmente mais relevante, é a relação ***acmB:courseUnits*** que ligando directamente *course* a *acmA:unit* do BoK ACM-A, permite indicar quais as unidades temáticas que serão abordadas na disciplina. Como referência para o restante texto e modelo, note-se que uma relação directamente com *acmA:unit* (conforme indicação no CC2001), sugere que o tema da *unit* é abordado na sua totalidade na disciplina, algo que pode não ser totalmente correcto e considera-se um aspecto a rever no ACM-B. Finalmente a relação opcional *acmB:coursePreReq*, que permite indicar quais os pré-requisitos existentes na disciplina através de 3 tipos de ligações: 1) para outras disciplinas (ou seja, conceito relaciona-se consigo próprio); 2) para *units*, indicando que é pré-requisito ter conhecimento dos temas de determinadas *units*; 3) e por fim, para o conceito definido por ***acmB:Background***, que indica como pré-requisito ter conhecimento (ou "*background*") sobre determinado assunto genérico, que não esteja modelado no corpo de conhecimento (ou seja, que não exista como *area*, *unit* ou *topic*). Note-se que, no âmbito unicamente da recolha da informação ACM sobre *course*s, este conceito simplesmente permitirá a identificação textual de temas não existentes no corpo de conhecimento (mas necessários como pré-requisitos). No entanto, considere-se como alerta que a sua eventual utilização para uma futura definição de unidades curricular poderá gerar um conjunto de informação necessário gerir e manter, que eventualmente poderá ser necessário "rever" para promoção para o corpo de conhecimento. Como última relação temos *acmB:courseSyllabus*, onde é efectuada uma ligação a elementos do tipo ***acmB:Syllabus****,* um para cada *course*, permitindo assim definir o programa ou *syllabus* proposto para a disciplina. O elemento *acmB:Syllabus* será por sua vez constituído pelos vários elementos constituintes do *syllabus*, modelados como uma lista de atributos textuais do tipo ***acmB:courseSyllabusTopic***. Note-se

que, seguindo a recomendação ACM e a definição dos vários *course*s do apêndice B do CC2001, estes atributos não são *topics* ACM-A mas sim elementos puramente textuais (na teoria, criados no momento de definição de *course*), que descrevem o programa definindo de forma textual os vários tópicos a abordar. A modelação deste aspecto seguindo a recomendação ACM, de imediato gera uma questão relativamente ao formato textual, pois através da relação *acmB:courseUnits*, e da identificação das *units* a abordar na disciplina, se poderá obter por inerência os seus *topics* ACM respectivos. Note-se que não existe efectivamente no modelo ACM qualquer relação entre *acmB:courseSyllabusTopic* e *acmA:topic*, além do seu eventual possível mesmo significado semântico.

3.3.3 CdC - *Skill*

Conforme o nome indica, esta componente do CdC terá a representação do conceito *skill* e seus sub-conceitos no modelo, de acordo com o apresentado na Figura 3.8, servindo assim como um repositório do corpo de conhecimento de competências necessário para a definição curricular.

Esta componente tem como principal elemento o conceito representado pelo objecto *di:skill*, aqui descrito, que sendo global ao sistema poderá ser utilizado (entenda-se "ligado") a todos os outros conceitos, quando for necessário a identificação de *skills* ou competências. Assim, considera-se para efeitos do modelo, que existirão *skills* com influência no saber, e portanto conhecimento ou "*knowledge"* sobre algo, com influência no fazer, e portanto a aplicação do conhecimento sobre algo, ou com influência num âmbito mais da personalidade, do social, hábitos, etc., e portanto conhecidas como *soft-skills*.

Os atributos directos do conceito são os seguintes: *i*) ***skillDescription***, que permite uma descrição textual da *skill*, com o detalhe necessário para que a mesma seja correctamente entendida. Como exemplo, veja-se uma *skill* recolhida com base nos *Dublin Descriptors*: "*Is able (with supervision) to critically reflect on his or her own thinking, decision making, and acting and to adjust these on the basis of this reflection.")*; *ii*) os atributos de classificação base ou genérica da *skill*, ***dK - Default Knowledge***, ***dS - Default Skill***, ***dA - Default Attitude***, onde é possível indicar se a skill terá, respectivamente, um ênfase no conhecimento e no "saber" do indivíduo, um ênfase mais prático e técnico, ou seja, nas características práticas do "fazer" do indivíduo, ou um ênfase maior sobre a "atitude" do indivíduo, portanto influenciando aspectos ou características referidas como "sociais" (tipicamente, definida como soft-skill). Note-se que, sendo o conceito *di:skill* genérico e aplicável a todo o sistema, os atributos d*K,dS,dA* são apenas indicadores de como a *skill*, na sua forma mais **genérica** (e tipicamente), influencia das diversas formas o indivíduo, podendo efectivamente variar de acordo como for abordada a sua aprendizagem. Daí o facto dos atributos serem descritos como "*default*" (ou seja, classificação por omissão ou na falta de nova indicação).

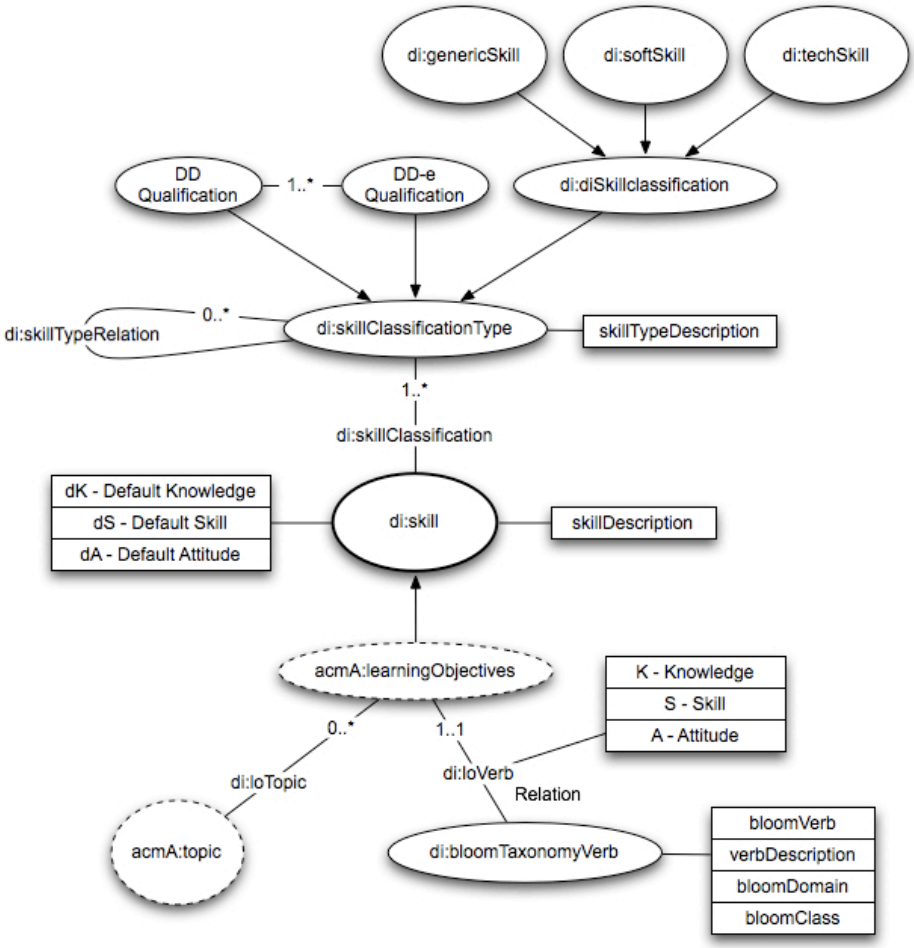


Figura 3.8 – Detalhe do modelo da componente CdC - *Skill*

Relativamente às relações, identificam-se duas, uma com o conceito de classificação da *skill* e outra para extensão da *skill*, utilizada pelo *acmA:learningObjectives*. Tendo já descrito anteriormente diferentes tipos de *skills*, distintos pela forma como influenciam as características do indivíduo, é natural assumir que estas deverão igualmente ser classificadas de várias formas, além da simples distribuição *dK*, *dS*, *dA* referida atrás. Assim, foi definida uma relação, de nome ***di:skillClassification***, que estabelece ligações a objectos do tipo *di:skillClassificationType* (descrito e justificado em seguida), que permite classificar a *skill* de diversas formas (ou indicar os seus tipo). Por fim, conforme referido atrás na descrição do conceito *acmA:learningObjectives*, os objectivos de aprendizagem são efectivamente definidos no sistema como uma **extensão** do conceito *skill*. Estes são na realidade, competências adquiridas pela aprendizagem de determinadas unidades temáticas ACM. Note-se que, sendo objectivos de aprendizagem, deverão ser classificados como *skills* de um teor técnico específico (independentemente da sua classificação dK,dS,dA), pois representam objectivos de temáticas ACM respeitantes à área da computação, sendo aquilo que se consideram como *skills* técnicas.

Passaremos então à descrição em detalhe dos conceitos relacionados, iniciando-se pela classificação de *skills*, conforme a Figura 3.8. Tendo por base a

necessidade de uma forma de classificação de *skill*s, anteriormente referido na relação *di:skillClassification*, que permita lidar com os múltiplos tipos de classificação, bem como com a evolução futura das mesmas, foi definido um objecto que representa o conceito de tipo de classificação de skill, nomeado ***di:skillClassificationType***, que deverá ser relacionado com o objecto *di:skill*. Este objecto deverá ser genérico o suficiente, para lidar com a futura criação no sistema (apenas pelos gestores do mesmo) de qualquer tipo de classificação ou agrupamento que se considere necessário para as *skills*. Assim, assumindo a necessidade da extensibilidade, optou-se por definir o objecto base *di:skillClassificationType* que deverá ser encarado como um objecto abstracto, do qual qualquer tipo de classificação deverá derivar. Isto permitirá a criação de diversos tipos, inclusive hierarquias de tipos, garantindo que poderão ser utilizados na classificação de qualquer *skill*.

Naturalmente, o objecto base *di:skillClassificationType* poderá já conter alguns atributos e relações úteis a qualquer tipo de classificação de *skill*. Primeiramente, o atributo ***skillTypeDescription****, que* conforme o nome indica, conterá a descrição do tipo de classificação. Adicionalmente, considerou-se que poderá existir a necessidade de relacionar os tipos de classificações entre si, não apenas hierarquicamente (ou seja, um subtipo), mas também com cruzamentos entre diversos níveis, sendo para isso definida relação ***di:skillTypeRelation***. Esta permite estabelecer ligações entre qualquer tipo de classificação, facilitando assim a indicação de relacionamentos (conforme necessário) entre qualquer dos tipos definidos. Note-se que se optou por não introduzir forma de descrever as relações (e distingui-las), pois o objectivo seria apenas a indicação de relação entre os tipos, devendo ser a semântica das mesmas conhecida pelos utilizadores no sistema.

Com recurso à extensão do objecto abstracto *skillClassificationType* considerou-se como base no sistema 3 tipos de classificação: *i*) Uma classificação da *skill* definida segundo **critérios do DI**, referidos na secção 2.5, representada no objecto *di:diSkillClassification*, sendo posteriormente dividida em 3 tipos distintos, *skill* genéricas (objecto *di:genericSkill*), *soft-skills* (no objecto *di:softSkill*) e *skills* técnicas (*di:techSkill*); *ii*) Uma classificação ***Dublin Descriptors,*** representada no objecto *DD_Qualification*; *iii*) e uma classificação **Dublin Descriptors da área da engenharia,** representada no objecto *DD-e_Qualification*. Far-se-á então uma descrição em detalhe cada um dos subtipos indicados.

A classificação definida segundo critérios do DI, representada pelo objecto ***di:diSkillClassification***, será uma forma de criar uma divisão base das *skills* no sistema, que permite uma classificação inicial (segundo normas DI) de qualquer *skill* que exista no sistema. Qualquer outra classificação que se verifique necessária, poderá ser então um objecto "irmão" (entende-se ao lado ou ao mesmo nível) que *di:diSkillClassification*. Tendo presente que se está no domínio de uma área técnica, a divisão entre 3 subtipos da classificação DI permite

identificar a *skill* de acordo com aquilo que se poderá considerar como a sua principal característica. Assim, conforme já referido, independentemente da classificação dK, dS, dA, uma *skill* que seja tipicamente técnica (sendo do "saber" ou do "fazer") será do tipo ***di:techSkill***, uma *skill* de uma natureza mais social (da personalidade, etc.) será do tipo ***di:softSkill***, e por fim, uma *skill* que seja mais genérica, aplicável a diferentes domínios e âmbito (não sendo facilmente classificável como técnicos ou sociais), será do tipo ***di:genericSkill***. Esta divisão, não pretendendo cobrir todas as possíveis distinções, permite rapidamente uma identificação do tipo de *skill*, facilitando a sua utilização no sistema, quer para a introdução de *skills* na UC, quer para possíveis comparações e *outputs* do sistema. Note-se que, caso se considere detalhar adicionalmente este tipos, ou seja introduzir novos tipos, o mecanismo de subtipos e as respectivas hierarquias poderão ser úteis, garantindo compatibilidade com futuras evoluções. Como exemplo de utilização, os objectivos de aprendizagem das unidades ACM (representados pelo objecto *acmA:learningObjectives*), deverão quase sempre considerados como *skills* do tipo *di:techSkill* (excepto em casos que reflictam directamente *soft-skills*), pois sabemos representarem objectivos a desenvolver pelos alunos (sejam teóricos ou práticos, e portanto, do "saber" ou do "fazer"), das temáticas das unidade ACM, que são por sua vez, temáticas no âmbito da área da ciência da computação, uma área inerentemente técnica.

Segundo o referido na secção 2.4.1, relativa aos *Dublin Descriptors (DD)*, pretendeu-se disponibilizar a possibilidade de associar qualquer *skill* que exista genericamente no sistema, à respectiva área de classificação *Dublin Descriptor* e assim estabelecer uma base comparativa com as competências de outros planos curriculares (por exemplo doutras universidades) que utilizem igualmente esta classificação. Revendo o descrito na secção 2.4.1, os descritores genéricos, orientados à identificação de competências genéricas para alunos que frequentem uma educação universitária, consideram 5 áreas de competência ou 5 descritores: 1) *Knowledge and understanding*; 2) *Applying knowledge and understanding*; 3) *Making judgements and decision making*; 4) *Communication*; 5) *Learning and self-learning skills*. Assim, conforme já referido acima, para suporte do conceito de descritor de *Dublin* genérico no sistema definiu-se o objecto ***DD_Qualification*** (ou seja uma qualificação DD de *skill*), utilizando-se novamente o mecanismo de subtipagem para declarar subtipos distintos para cada um dos 5 *DD*, identificado cada um com o seu número respectivo.

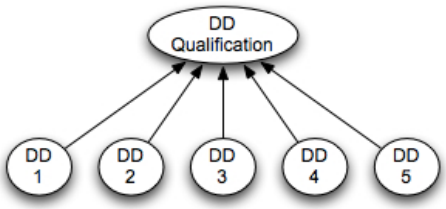


Figura 3.9 - O conceito *DD_Qualification* e respectivos descritores

Terminando a descrição dos vários tipos de *skills*, falta apenas o referido na secção 2.4.2, ou seja, os *Dublin Descriptores Engineering* (DD-e). Em particular, pelo facto do âmbito desta dissertação ser o DI, um departamento que lecciona ciência da computação, faz sentido considerar estes descritores mais focados na área de engenharia, específicos para o 1º e 2º ciclo de Bolonha. Para representação destes DD-e considerou-se o objecto ***DD-e_Qualification***, representado na Figura 3.10, que conforme os DD atrás referidos, estabelece também uma possível base comparativa com outros cursos e outras universidades que utilizem na sua representação de skills uma classificação deste tipo. Relembrando o referido na secção 2.4.2, os descritores de Dublin para engenharia (DD-e) definem 7 áreas ou divisões principais, relativas ao tipo de competência ou *skill*: A) *Competent in one or more scientific disciplines*; B) *Competent in doing research*; C) *Competent in designing*; D) *A scientific approach*; E) *Basic intellectual skills*; F) *Competent in co-operating and communicating*; G) *Takes account of the temporal and social context*; De forma similar aos DD, a representação dos 7 *DD-e* no sistema faz igualmente uso da extensibilidade ou subtipagem, declarando subtipos para cada um dos *DD-e*, permitindo que estes sejam relacionados directamente com *skills* (via a relação *di:skillClassification*) ou referenciados por outros objectos, quando necessário.

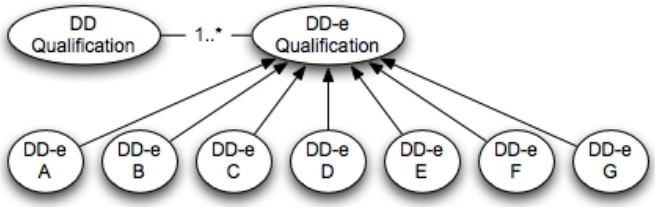


Figura 3.10 - O conceito *DD-e_Qualification* e respectivos descritores

Chama-se a atenção para a relação entre os objectos *DD-e_Qualification* e *DD_Qualification*, que pretende responder à tabela de correspondência entre DD e DD-e, definida na Tabela 2.2 da secção 2.5 (onde se aborda a proposta do DI de adequação a Bolonha). Esta correspondência foi mapeada com recurso à relação *di:skillTypeRelation* (existente entre qualquer tipo de classificação de *skill*), sendo assim possível partir de qualquer DD-e indicar os respectivos DD. Isto permite melhorar a definição e possibilidades comparativas das competências e *skills* definidas no modelo.

Uma nota final sobre os DD-e é o facto do trabalho que os propõe, "*Criteria for Academic Bachelor's and Master's Curricula*" [73], indicar em cada DD-e um conjunto de *skills* genéricas que deverão ser adquiridas quer pelos alunos do 1º ciclo, quer pelos alunos do 2º ciclo (refere-se a Figura 2.29). Pelo âmbito relevante e genérico deste trabalho, focado na engenharia e utilizável em várias universidades ou vários cursos, as *skills* propostas são relevantes o suficiente para serem introduzidas também neste sistema. Estas *skills* estão detalhadas a um nível mais alto de representação das competências a adquirir (um nível

menos técnico e mais orientado a competências de ordem científica geral e intelectual), o que poderá ser também útil na especificação mais genérica de *skills* fornecidas pelas UC. Assim, optou-se por considerar na fase de povoamento do modelo a possível introdução futura destas *skills* no sistema, sendo definida para cada uma a classificação com o respectivo descritor DD-e e a classificação *di:genericSkill*.

Para terminar o detalhe do conceito *di:Skill* falta apenas elaborar sobre a extensão pelo conceito **acmA:learningObjectives** (relembrar a Figura 3.8). O facto destes objectivos serem considerados igualmente *skills* obriga naturalmente a pensar na remodelação da forma, eventualmente simplista, com que foram originalmente definidos nos conceitos ACM-A. Para isto, optou-se por fazer uma análise em maior detalhe dos textos ACM-A dos objectivos de aprendizagem e da forma como estes poderiam conter informação que desse pistas para o preenchimento de algumas das características já definidas das *skills* ou eventualmente potenciar eventuais remodelações do modelo das mesmas. Analisando qualquer das *unit* ACM e seus componentes, identifica-se um padrão de relacionamento e correspondência entre cada um dos tópicos - o detalhe da temática a abordar na unidade; e cada um dos objectivos de aprendizagem - as competências adquiridas após aprendizagem da temática da unidade. Apesar de expectável este relacionamento, tal não é estipulado pelo *BoK* ACM-A. Aliás, verifica-se que, tipicamente em quase todos os objectivos de aprendizagem, o texto que o descreve está semanticamente relacionado (e por vezes, quase palavra por palavra), com um tópico específico (ou mais do que um) da mesma unidade, sendo geralmente precedido por uma acção que descreve genericamente a *skill* que se pretende desenvolvida sobre esse tema. Veja-se um exemplo concreto na Figura 3.11, onde com base na leitura e interpretação dos textos da *unit* ACM-A "PF3", é possível estabelecer a tabela abaixo.

Executando este tipo de análise comparativa para várias unidades do *BoK* ACM-A é possível verificar que, além da correspondência habitual entre os *LO's* e *Topics*, as *skills* descritas são tipicamente precedidas sempre por um verbo ou "acção" pertencente a um conjunto reduzido. Referindo a secção 2.4.3, verificou-se que estes verbos estão incluídos nas palavras chave da taxonomia de *Bloom*, dentro dos 3 domínios principais: cognitivo (aprendizagem e competências "mentais"), afectivo (aprendizagem da área emocional e social) e psicomotor (competências manuais e físicas). Pode-se também dizer que estes domínios principais são equivalentes às classificações K,S,A (*Knowledge, Skills, and Attitude*), já descritos na *skill*, que indicam a forma de influência da mesma no indivíduo. Assim, este verbo ou "acção" principal do objectivo de aprendizagem, identifica aquilo que tipicamente classifica o tipo de *skill* e a forma como esta influencia o indivíduo.

PL3. Introduction to language translation [core]

Minimum core coverage time: 2 hours

Topics:

1 Comparison of interpreters and compilers
2 Language translation phases (lexical analysis, parsing, code generation, optimization)
3 Machine-dependent and machine-independent aspects of translation

Learning objectives:

1. Compare and contrast compiled and interpreted execution models, outlining the relative merits of each..
2. Describe the phases of program translation from source code to executable code and the files produced by these phases.
3. Explain the differences between machine-dependent and machine-independent translation and where these differences are evident in the translation process.

LO	***Topic***	***Skill* descrita no *LO***	***Skill* – acção**
1	1	*Compare and contrast [...], outlining the relative merits of each.*	*Compare and contrast*
2	2	*Describe [...]*	*Describe*
3	3	*Explain the differences [...] and where these differences are evident [...].*	*Explain the differences*

Figura 3.11 – Correspondências entre *LO's* e *Topics* da *unit* ACM-A "PF3"

Com base nesta análise, e relativamente ao proposto pelo ACM-A, é possível propor um modelo melhorado de acmA:learningObectives, onde se estabelece que: *i*) São *skill*s, e portando sendo extensões de *di:Skill*, utilizam qualquer dos seus atributos/relações e são "utilizáveis" genericamente no modelo; *ii*) Existe uma relação entre *LO's* e *Topics*; *iii*) E existe uma possível identificação da classificação das *skills* definidas pelos *LO's*, tendo por base os seus textos. Considera-se que esta proposta poderá ser útil em futuros trabalhos sobre os conceitos ACM.

Na Figura 3.12, identificam-se então nos LO's 2 relações adicionais: *i*) A relação *di:loTopic,* que permite estabelecer uma ligação aos *acmA:topic* que são descritos como competência adquirida (note-se que poderão existir *LO's* que se liguem a vários *Topics*, ou LO's que não se consigam ligar a nada relevante); *ii*) A relação *di:loVerb*, que estabelece a identificação de que verbo da taxonomia de *Bloom* é referido no objectivo de aprendizagem, com uma ligação ao conceito *di:bloomTaxonomyVerb* (descrito adiante). Com base no verbo ou "acção" extraído do texto LO's, é possível ainda reclassificar a forma como esta competência influencia o indivíduo, recorrendo aos típicos atributos K,S,A, já referidos na *skill*. Note-se que a classificação K,S,A existente nesta relação, apenas se refere à utilização do verbo **neste** objectivo de aprendizagem (podendo naturalmente variar noutros LO's), podendo no entanto o sugerir o preenchimento eventualmente automático dos atributos dK,dS,dA do LO's.

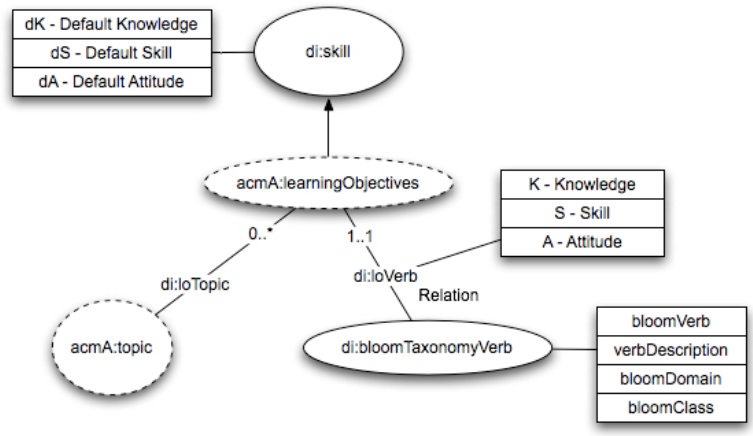


Figura 3.12 – Extensão e remodelação dos *acmA:learningObjectives*

Por fim, terminando a descrição desta proposta de melhoria, apresenta-se o conceito ***di:bloomTaxonomyVerb*** como forma como forma de representar os verbos da taxonomia de *Bloom* no modelo (considere-se que será um objecto global desta secção CdC-SKILL, e portanto utilizável para outros conceitos no modelo). Este é constituído pelos seguintes atributos: *i*) *bloomVerb,* que indica textualmente numa palavra, sob a forma sintáctica simples apresentada na taxonomia de Bloom, qual o verbo retratado; *ii*) *verbDescription,* uma descrição textual detalhada do que se pretende indicar com o verbo declarado, de forma a evitar ambiguidades entre os diversos verbos; *iii*) *bloomDomain*, uma indicação do domínio do verbo, segundo definido na taxonomia de Bloom: cognitivo, afectivo, psicomotor. Note-se que o mesmo verbo poder ser utilizado em vários domínios, sendo então o par verbo-domínio único (entenda-se "chave primária"); *iv*) A indicação da classe do verbo dentro do domínio, de acordo com o definido na taxonomia (refere-se à Figura 2.32 da descrição da taxonomia de *Bloom*).

3.3.4 A Unidade Curricular - UC

Passar-se-á então à descrição dos conceitos relacionados com a Unidade Curricular e sua instância, dois dos conceitos mais relevantes no modelo.

A utilização do conceito *acmB:course* para construção das UC deverá ser entendido como uma "extensão" do conceito, artefacto já utilizado e referido anteriormente (similar aos conceitos de herança nas linguagens de programação [95]).

Conforme se pode verificar na Figura 3.13, este é aplicável às UC, permitindo a reutilização e redefinição dos seus atributos e relações. Esta extensão, além de permitir a reutilização das relações / atributos de *acmB:course* (ou seja, número, título, descrição, notas, *Syllabus*, ACM *units*, pré-requisitos, etc.), permite também identificar as unidades curriculares (UC) como sendo elementos *acmB:course*, o que poderá ser útil para uma eventual possibilidade de comparação com outros modelos de outras universidades. A mesma lógica é aplicada às instâncias de UC, que "estendem" o conceito UC, reutilizando e redefinindo os seus atributos e relações (e por inerência os de *acmB:course*).

Existe ainda uma relação *cuModel*, que permite indicar qual UC (modelo) foi utilizando para a criação/definição da instância de UC.

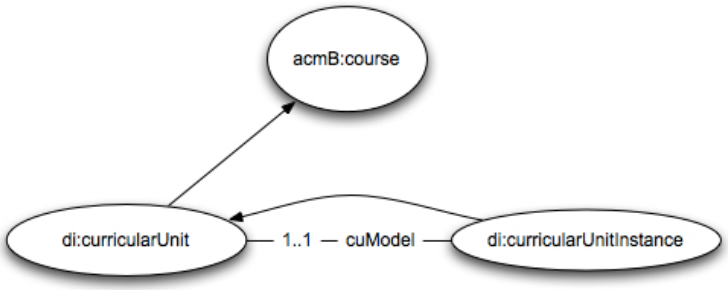


Figura 3.13 – UC e instância UC com base no conceito *acmB:course*

Como forma construir um modelo coerente com o conjunto de informação já actualmente existente nas disciplinas do DI (refere-se a informação do documento de remodelação de Bolonha do curso de Eng.ª Informática do DI [73]), as UC baseadas na extensão de *acmB:course* necessitam adicionalmente dos seguintes componentes genéricos: *i*) **Atributos adicionais** do departamento e da faculdade para cada UC (tipo da cadeira - opcional ou obrigatório, sigla da cadeira, etc.); *ii*) **Objectivos de aprendizagem**, ou competências / qualificações / *skills* que o aluno deverá ter adquirido após frequentar a UC (do SABER, do FAZER e *Soft-Skills*); *iii*) **ECTS** da UC; *iv*) **Esforço do aluno**, relativamente às horas envolvidas na aprendizagem pelo aluno das temáticas da UC; *v*) **Bibliografia** de apoio à temática da UC; *vi*) **Metodologia da Avaliação** da UC; *vii*) **Pré-Requisitos recomendados** (pois os obrigatórios já existem no conceito *acmB:course*) para frequência da UC.

Pode-se então apresentar a Tabela 3.1, que descreve em alto nível os vários componentes de informação que se julgam necessários, sendo descritos posteriormente em maior detalhe a quando da apresentação dos vários conceitos. Os componentes foram indicados tendo em consideração a sua reutilização com base no CdC ACM-B ou criação de raiz, e indicação onde serão primeiramente declarados / definidos. É importante ter presente a distinção entre UC e instância de UC: A primeira deverá conter **aspectos relacionados com a organização temática e conceptual**, enquanto que a segunda deverá conter **aspectos relacionados com a "execução" operacional** efectiva, **sendo no entanto possível alterar** (nas suas operacionalizações anuais) **as definições conceptuais da UC que lhe deram origem** (ou seja, as instâncias reutilizam e poderão alterar).

	Origem		**Pertence a**		
Componente	**acmB**	**Novo**	**UC**	**Inst.**	**Descrição**
Pré-requisitos da UC	√		√		Considerando efectivamente os 3 tipos já definidos de pré-requisitos, acmB:course (e portanto UC), acmA:unit e acmB:background.
Lista das Unidades ACM	Precisa ajuste		√		As unidades necessitam ser qualificadas ("pesadas") por ECTS, com

	Origem		Pertence a		
Componente	**acmB**	**Novo**	**UC**	**Inst.**	**Descrição**
(acmA:unit) abordadas na UC (temática)					respectiva divisão do esforço por aulas, estudo, etc., conforme actual definição no DI.
Programa da UC	Precisa ajuste		√	√ (ext.)	Considerou-se a extensão de acmB:courseSyllabus, com uma possível uma organização em secções de tópicos a serem abordados. Os tópicos ser os elementos acmA:topic (baseados nas unidades acima) ou os objectos textuais descritos por acmB:courseSyllabusTopic. Note-se que deverá ser um elemento da UC, com uma extensão/ ajuste na sua instância, devido à definição operacional necessária
ECTS da UC		√	√		Total à cadeira, este deverá ser utilizado para "pesos" (distribuição de relevância) dos diversos elementos da UC, quantificando o esforço do aluno.
Esforço do aluno		√			Qualificado de diversos tipos (aulas teóricas, práticas, etc.), habitualmente declarado como horas.
Pré-requisitos recomendados		√	√		Considerando efectivamente a mesma estrutura de pré-requisitos, mas terão apenas um carácter de recomendação e não de obrigação (ao contrário dos acima descritos).
Bibliografia		√		√	Considerando a questão de escalabilidade e extensão do sistema, optou-se por uma abordagem genérica a materiais de referência, que poderão ser estendidos ou sub-tipados futuramente.
Avaliação na UC		√		√	O conceito permitirá definir qual o formato de avaliação na UC: qual o seu tipo, qual o seu peso, quais os tópicos abordados, quais as competências que o aluno deve atingir), etc.. Note-se que pretende-se no modelo apenas especificar a avaliação, não os resultados dos alunos. Os resultados poderão eventualmente ser cruzados globalmente com recurso a outro sistema.
Formas de ensino na UC		√		√	Onde serão as formas de ensino utilizadas na UC (ex: aulas teóricas, seminários, etc.): o seu tipo e descrição, a sua relação com a avaliação, as competências que se pretende que os alunos atinjam etc.

	Origem		Pertence a		
Componente	**acmB**	**Novo**	**UC**	**Inst.**	**Descrição**
Competências adquiridas na aprendizagem da UC		√	√		Conceito que deverá ser integrado com os diversos componentes da UC, e que representa as qualificações (no sentido lato), que deverão ser adquiridas pelos alunos após a frequência da UC (ou de determinado do componente da UC). São essencialmente os "objectivos" da UC. As competências deverão ser uma base comparativa importante, quer entre UC, quer entre estruturas curriculares, quer entre currículos de universidades. As qualificações deverão ter em conta os conceitos anteriormente referidos de di:Skill, permitindo assim uma fácil especificação por parte de quem declara a UC.

Tabela 3.1 - Elementos propostos para a UC e instância de UC

Com base nesta visão inicial, passa-se então à descrição do conceito UC, que é representado no modelo pelo objecto ***di:curricularUnit***. A UC, conforme já descrito, deverá permitir uma definição das principais componentes de organização temática / objectivos, com recurso a relações para outros conceitos, não deixando por isso de necessitar também de alguns atributos mais genéricos de identificação e controlo.

3.3.4.1 O conceito *di:curricularUnit*

Pelo facto do modelo de UC ser uma "extensão" do conceito *acmB:course*, é possível utilizar qualquer dos atributos e relações já definidos, portanto só estes só serão referidos quando for necessário rever algum aspecto. Por exemplo, dado as especificidades que serão necessárias incluir na forma de definição das temáticas da UC, definiram-se duas relações novas, *di:cuAcmUnit* e *di:cuSyllabus*, que reescrevem e permitem ignorar as relações originalmente estendidas do conceito *acmB:course*, *acmB:courseUnits* e *acmB:courseSyllabus*. Note-se que, apesar de se considerar estas relações herdadas como "ignoradas" ou opcionais, pode-se eventualmente fazer o seu preenchimento (se possível) com a informação de base estabelecida nas novas relações, garantindo assim uma coerência na disponibilização das UC como *acmB:course*, para possível cruzamento com fontes externas.

Veja-se então Figura 3.14, onde se apresenta o conceito *di:CurricularUnit* com os seus atributos e relações, que se descrevem em seguida. Como atributos do objecto que representa UC consideram-se: *i*) ***di:cuAlias****,* uma sigla ou outra designação alternativa para a UC. É um atributo textual, obrigatório e único no sistema. *ii*) ***di:cuECTS****,* representa o valor ECTS da UC, que quantifica o esforço do aluno para aprendizagem da UC. Este valor deverá ser sempre indicado manualmente em cada UC, sendo que qualquer cálculo de ECTS necessário para outros conceitos da UC, deverá utilizar este como base. Conforme o conceito ECTS, este valor numérico poderá ser convertido no valor de "esforço médio em

horas" para aprendizagem da UC pelo aluno, com base no valor fixo "horas por ECTS" estipulado geralmente pelas faculdade; *iii*) ***di:Version***, representa a versão de determinada UC. Este atributo, em conjunto com o atributo herdado *acmB:courseNumber* (número da disciplina), serão a "chave" da UC <*acmB:courseNumber*, *di:Version*>.

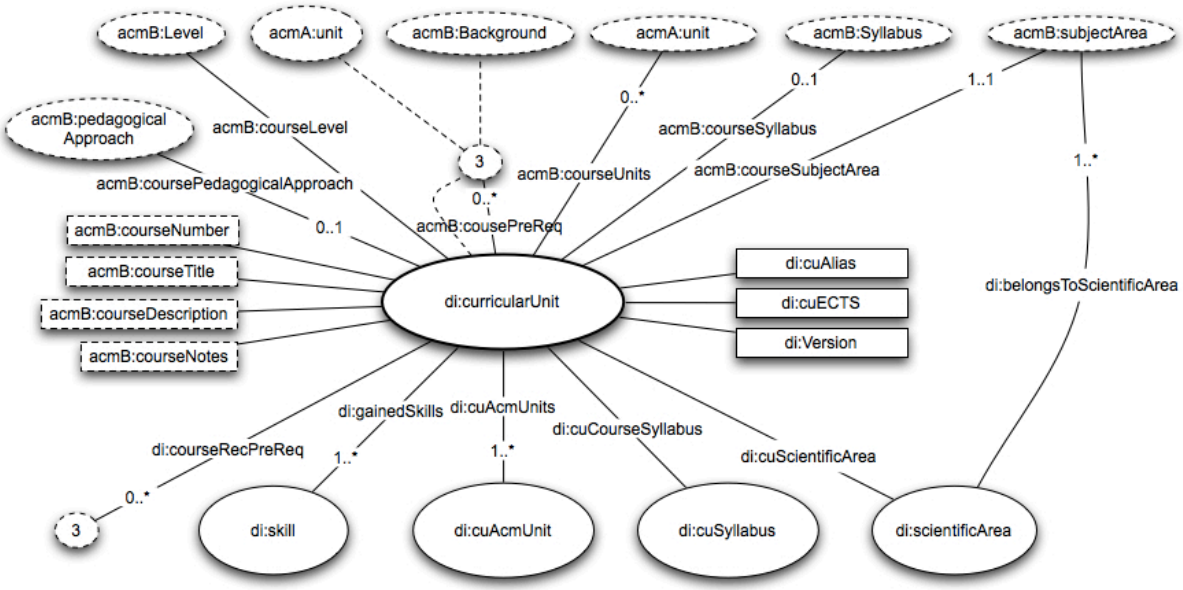


Figura 3.14 - O conceito *di:curricularUnit*

As relações, porém, são mais complexas e permitem a efectiva definição conceptual da UC. Iniciaremos pelas relações herdadas de *acmB:course*, às quais consideraremos modificações: *i*) ***acmB:courseSyllabus***, será opcional; *ii*) ***acmB:courseUnits***, será opcional; As novas relações propostas, existem então para responder às novas componentes necessárias na UC: *iii*) ***di:cuAcmUnits***, representa uma ligação entre a UC e o conceito que identifica as temáticas a abordar na disciplina. Utilizar apenas uma ligação ao conceito *acmA:units*, não é suficiente para uma representação total do conhecimento que se pretende introduzir na descrição da temática a abordar na UC, por este motivo, foi necessário especificar um elemento intermédio na relação entre a UC e as *acmA:units*, de nome *di:cuAcmUnit*, que está descrito em seguida. Note-se que só poderá existir UMA relação da UC para um elemento *di:cuAcmUnit* com determinado *acmA:unit* (ou seja, uma temática *acmA:unit* por UC). Esta relação, conforme já referido deverá manter a coerência com a relação opcional *acmB:courseUnits*; *iv*) ***di:cuCourseSyllabus***, que representa uma ligação ao conceito *di:cuSyllabus*, descrito mais à frente, onde se permite a definição do *syllabus*[21] da UC. Este será descrito com recurso aos *acmA:topic* que fazem parte

[21] *Syllabus* – Do inglês: um esboço ou sumário dos principais pontos de um texto, apresentação, seminário ou disciplina de estudo.

das unidades a abordar na UC. A ordem de apresentação dos *topics* é irrelevante na definição da UC (mas não o é na instância); *v*) ***di:gainedSkills***, que representa uma ligação entre a UC e as competências que deverão ser adquiridas pelos alunos após a frequência da UC, com recurso ao conceito de *di:skill* (atrás referido). A relação *di:gainedSkills* abordar novamente mais à frente, pois contém ela própria atributos que classificam a *skill* na sua utilização da UC; *vi*) ***di:courseRecPreReq***, que representa uma ligação opcional, similar à relação *acmB:coursePreReq*, onde se identificam os pré-requisitos da disciplina, mas com teor de recomendação e portanto não obrigatórios. Segue a lógica da relação *acmB:coursePreReq*, permitindo ligações para UC (que são também *acmB:course*), *acmA:unit* e "Background"; *vii*) ***di:cuScientificArea***, que representa a área científica da disciplina, estabelecendo uma relação ao conceito *di:scientificArea*, que mapeia a noção de área científica no modelo e será descrito mais à frente. Note-se a relação entre este conceito *e acmB:subjectArea*, que permite indicar que determinada área científica pode conter vários *acmB:subjectArea*.

Tendo esta visão global da UC é possível então passar ao detalhe dos vários conceitos referenciados, iniciando pelo ***di:cuAcmUnit***. Veja-se a Figura 3.15.

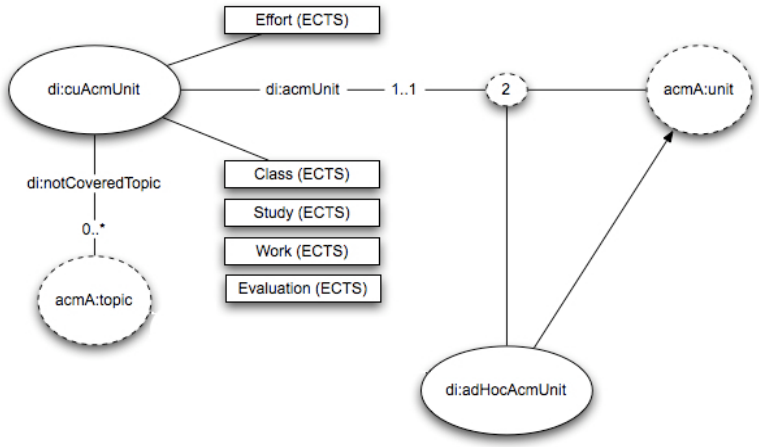


Figura 3.15 - *di:cuAcmUnit*, ligações e restantes componentes

Conforme descrito acima, considerar na UC apenas uma ligação ao conceito *acmA:units* não é suficiente para os objectivos da UC, tendo sido por isso especificado o conceito intermédio *di:cuAcmUnit* na relação entre a UC e as *acmA:units*. Este elemento deverá efectivamente representar uma extensão "conceptual" (e não herança) do conceito das *acmA:unit* dentro do modelo, que seja aplicável às UC e restante componentes. Os elementos *di:cuAcmUnit* serão, em diante, considerados como as **unidades da UC**, dado que representam as unidades temáticas a abordar na UC.

O objectivo da criação deste elemento, é permitir definir na UC os seguintes aspectos adicionais, que aumentam o detalhe de informação da UC: *i*)

Importância / **relevância** (ou mais simplesmente, o "peso") de cada *acmA:Unit* a abordar na UC; *ii*) Dado que *acmA:unit* agrupa um conjunto de tópicos do mesmo tema, é provável que surja a necessidade de não os abordar todos na UC, mesmo sendo grande parte da *unit* coberta. É então útil a possibilidade de especificar ***acmA:topic* que não são abordados na temática da UC**, apesar de serem parte dessa *acmA:unit*; *iii*) Inversamente ao ponto anterior, poderá ser necessário abordar apenas um, ou um conjunto muito reduzido de *acmA:topic* distribuídos por várias *acmA:unit*, sendo pouco prático escolher uma *unit* e depois indicar que todos os *topic*, excepto poucos, não serão abordados. Foi definido então um **elemento que permita agrupar *topics* individuais**, que não tenham ainda presença noutra *unit*, de forma similar a uma *unit* especificada *ad-hoc* no momento de definição da UC.

Assim, com base nos objectivos, é possível detalhar os atributos e relações do conceito. Iniciando-se pelos atributos: *i*) ***Effort***, representa a classificação do peso, importância ou relevância, de cada uma das unidades (di:cuAcmUnit) dentro da UC. Dado que esta "importância" pode ter vários significados, inclusive subjectivos, define-se que a dificuldade de aprendizagem, ou esforço envolvido na aprendizagem da unidade pelo aluno, será o valor de "relevância" a utilizar. Esta medida de esforço do aluno são os ECTS da UC, o que poderá ser utilizado para a distribuição de "peso" entre as diversas unidades. Assim, o atributo "*Effort (ECTS)*", deverá ser o esforço de aprendizagem da unidade na UC à qual pertence, expresso em ECTS (seja valor ECTS ou horas), tendo como base o valor de ECTS da UC, que será distribuído por inteiro entre as unidades da UC definidas; *ii*) Seguindo o raciocínio do esforço de aprendizagem pelo aluno e respectiva distribuição dos ECTS pelas unidades da UC, verificou-se fazer sentido ter esses esforço quantificado em maior detalhe, conforme já definido nas disciplinas actualmente do DI, que consideram o esforço do aluno (em horas / ECTS) dividido por aulas teóricas, práticas, estudo, etc.. Consideram-se então os atributos ***Class, Study, Work, Evaluation***, que são uma divisão simples dos diferentes tipos de "trabalho" do aluno dentro da unidade. Respectivamente, o esforço envolvido unicamente em aulas (no sentido genérico do conceito, teóricas, teórico-práticas, etc.), o esforço do relativo ao estudo (preparação para aulas, estudo para exames, etc.), o esforço do aluno sobre os trabalhos desenvolvidos (trabalhos práticos ou entregáveis que não exames para avaliação), e por fim o esforço do aluno relativamente à avaliação (testes, exames, chamadas orais, etc.). Em resumo, cada unidade da UC deverá ter uma parcela dos ECTS totais da UC, que por sua vez poderão ser distribuídos pelos 4 eixos da unidade, conforme as equações na Figura 3.16.

$$ECTS^{UC} = \sum_{n} ECTS^{UC}_{unidade\_n}$$

$$ECTS_{unidade\_n} = Class^{ECTS}_{unidade\_n} + Work^{ECTS}_{unidade\_n} + Study^{ECTS}_{unidade\_n} + Evaluation^{ECTS}_{unidade\_n}$$

Figura 3.16 – Distribuição das ECTS pelas unidades da UC

Passando às relações do conceito *di:cuAcmUnit*, definem-se as seguintes: *i*) a principal relação ***di:acmUnit****,* que permite indicar qual é a unidade temática que deverá ser "ligada" à UC por intermédio do conceito *di:cuAcmUnit*, representando assim uma das unidades a abordar. Conforme verificado na Figura 3.15, esta ligação é estabelecida para 2 tipos de conceitos, unidades ACM do tipo *acmA:unit*, ou para uma unidade AD-HOC criada no âmbito desta UC, a qual é mapeada no conceito *di:adHocAcmUnit* (descrito de seguida) onde se indicam de forma *ad-hoc* diversos *acmA:topic* a abordar na UC; *ii*) ***di:notConveredTopic***, que permite indicar se algum dos tópicos da unidade referida pela relação anterior (seja *acmA:unit* ou unidade *ad-hoc*), não será abordado nesta UC. É estabelecida uma relação que "liga" o conceito di:cuAcmUnit, a elementos acmA:topic que existem dentro da unidade referida pela relação *di:acmUnit*, indicando assim quais NÃO serão abordados na UC;

Para se finalizar o detalhe de *di:cuAcmUnit*, falta então detalhar o conceito ***di:adHocAcmUnit***, representado na Figura 3.17.

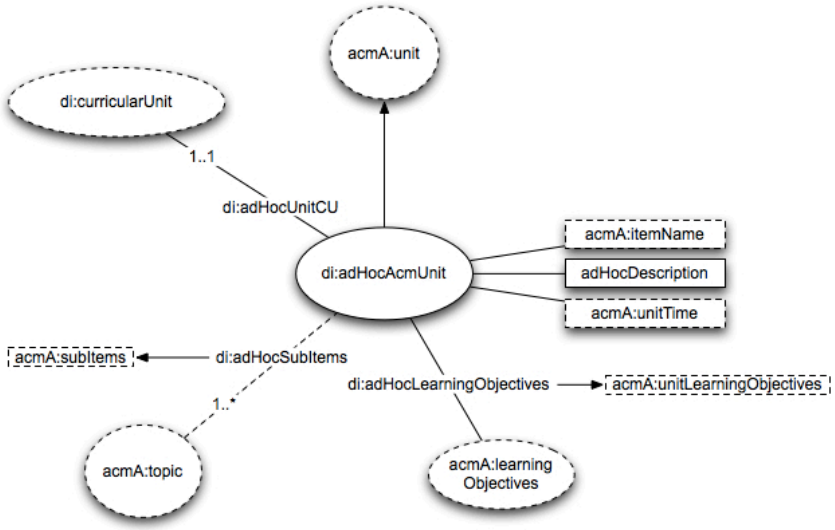


Figura 3.17 - O conceito *di:adHocAcmUnit*, atributos e relações

Este conceito, conforme referido atrás, tem por objectivo permitir que seja possível definir uma unidade de forma *AD-HOC* e escolher individualmente os tópicos a abordar, evitando que se tenham de especificar as várias unidades ACM e indicar que a maioria dos tópicos dessas unidades não serão abordados na UC. É importante reter que o âmbito deste tipo de elementos deve ser apenas e unicamente da UC onde foi definido, não sendo possível a sua reutilização noutras UC. Adicionalmente, e conforme a figura do conceito, de forma às unidades *ad-hoc* serem efectivamente consideradas como unidades temáticas (na acepção do conceito ACM), este conceito **estende** *acmA:unit* e aproveita / redefine os atributos e relações herdadas (veja-se a seta para o conceito *acmA:unit*).

A criação de uma unidade *ad-hoc* pressupõe que deverão ser preenchidos manualmente pelo especificador os seus detalhes, de acordo com os atributos e relações que se descrevem em seguida. Verifique-se que se reaproveitam os

atributos ***acmA:itemName*** e ***acmA:unitTime***, para especificar o nome da unidade ad-hoc e o tempo mínimo esperado para a sua abordagem (que se manterá opcional). Porém, considerou-se que o atributo ***acmA:unitType*** (*core* ou *elective*) não será utilizado nas unidades *ad-hoc*. Adicionalmente acrescenta-se um novo atributo, ***adHocDescription***, onde se indicará de forma textual detalhada a descrição da unidade *ad-hoc*, devendo ser focados os objectivos e o porquê da criação desta unidade *ad-hoc* com os tópicos escolhidos. Relativamente às relações, estende-se de *acmA:unit*, as relações *acmA:subItems* e *acmA:unitLearningObjectives*, criando assim as relações ***di:adHocSubItems*** e ***di:adHocLearningObjectives***. Estas permitem, respectivamente, indicar o conjunto de tópicos *acmA:topic* (com base nas diferentes *units* no CdC ACM-A) que esta unidade ad-hoc deverá abordar e indicar os objectivos de aprendizagem da unidade, com base nos tópicos definidos atrás (poderá não ser possível uma inferência automática com base nos tópicos). Adicionalmente acrescentou-se a relação ***di:adHocUnitCU***, que permite estabelecer uma ligação directa da unidade ad-hoc à UC. Considerando que as unidades *ad-hoc* existem originalmente apenas ligadas via conceito *di:cuAcmUnit*, pretende-se enfatizar que a unidade *ad-hoc* definida é especificada no âmbito daquela UC particular (naturalmente, o preenchimento desta relação deverá ser automático, dado que o utilizador está no contexto de determinada UC quando define a unidade *ad-hoc*). Por fim, a relação *acmA:subItemOf* (herdada da extensão acmA:unit) não será utilizada nas unidades *ad-hoc*, pois estas não estarão integradas na hierarquia ACM.

Finalizando a descrição do conceito *di:adHocAcmUnit*, refere-se um conjunto de regras inerente à ligação de tópicos da unidade *ad-hoc*. Só poderão ser indicados *acmA:topic* que: 1) "Pertençam" a uma *acmA:unit* que **não** esteja anteriormente referida na UC da qual a unidade *ad-hoc* pertence; 2) Ou, caso pertençam, deverão estar indicados como tópicos não cobertos, via relação *di:notConveredTopic* (no conceito *di:cuAcmUnit*).

Continuando a descrição dos conceitos associados a representação da UC, di:curricularUnit, passaremos então à descrição do conceito **di:cuSyllabus,** referido anteriormente na relação *di:cuCourseSyllabus* da UC e representado na Figura 3.18.

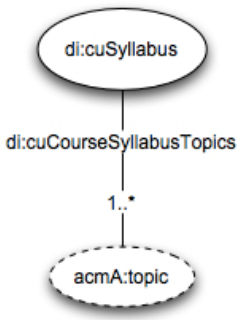


Figura 3.18 - o conceito *di:cuSyllabus* e relação para *acmA:topic*

Um syllabus deverá ser uma descrição detalhada, orientada aos constituintes simples da temática a abordar. Considerando que a UC contém um conjunto de unidades a abordar (sejam *acmA:unit* e *ad-hoc*) e que estas são constituídas em última instância por tópicos ACM, o *syllabus* poderá ser definido tendo por base os *acmA:topic* das respectivas unidades declaradas. Foi então definido um conceito *di:cuSyllabus*, existente unicamente no âmbito de *di:curricularUnit*, que estabelece ligações para os tópicos ACM (*acmA:topic*) das unidades previamente declaradas na UC, com recurso à relação ***di:cuCourseSyllabusTopics***. Os tópicos do *syllabus* serão um agrupamento automaticamente definido com base nos tópicos das unidades já declaradas na UC, incluindo unidades *ad-hoc*, não considerando naturalmente os tópicos indicados como não abordados (via relação *di:notConveredTopic*). Refere-se que, dado que a UC se foca com aspectos estruturais, não importa a ordem pelo quais os tópicos são indicados no *syllabus*.

Voltando à descrição dos conceitos relacionados com di:curricularUnit, passa-se à descrição em maior detalhe da relação ***di:gainedSkills***, que pretende indicar quais as competências ou *skills* que um aluno, obtendo frequência na UC, deverá ter adquirido. Conforme já referido, estas competências são indicadas com recurso ao conceito *di:Skill*, estabelecendo a relação representada na Figura 3.19.

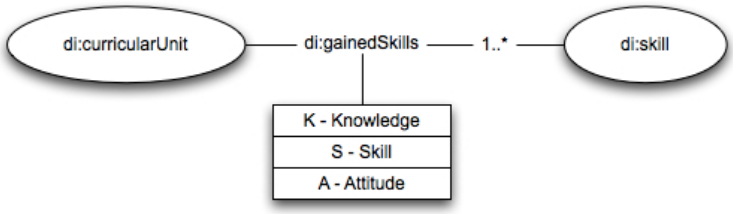


Figura 3.19 – As competências da UC - a relação di:gainedSkills

Para contextualização, refere-se o conceito *di:Skill*, onde se pode identificar qual a forma genérica / *default* (ou por omissão) como a *skill* influencia o indivíduo que a adquire, em particular os atributos dK, dS, dA. Da mesma forma, na indicação da *skill* proporcionada pela UC, se pretende estipular qual o ênfase dado ao ensino da mesma nesta UC. Permite-se assim novamente uma reclassificação, apenas no âmbito da UC, sobre os 3 mesmos eixos base (***Knowledge***, ***Skill*** e ***Attitude***), indicando quais deles (independentemente) reflectem o ênfase pretendido desta *skill* na UC.

Considere-se também que, o facto de se declararem unidades a abordar na UC, está igualmente a indicar *skills* alvo da UC, dado que existem objectivos de aprendizagem (*acmA:learningObjectives*) nas unidades declaradas. Assim, define-se que a especificação das *skills* na UC pode ser maioritariamente executada de forma automática, preenchendo esta relação com ligações aos *acmA:learningObjectives* especificados nas várias unidades da UC. Naturalmente haverá sempre a possibilidade de introdução manual de *skills* pelo especificador da UC, assumindo-se porém, que só poderão ser introduzidas *skills* que não existam já relacionadas. Igualmente, para o preenchimento dos atributos K, S, A,

se poderá considerar a automatização com base na informação genérica dos atributos dK, dS, dA, dos conceitos *di:Skill*, podendo sempre ser alterados de acordo com as especificidades da UC e o ênfase que se pretenderá à *skill* adquirida. Por fim, a remoção manual de uma *skill* que tenha sido automaticamente incluída com base nas unidades (portanto um objectivo de aprendizagem), deverá ter um aviso informativo de confirmação, pois os objectivos de aprendizagem estão naturalmente associados aos tópicos das unidades, só fazendo sentido remover caso o tópico não seja abordado.

Por fim, terminando a descrição dos conceitos relacionados com di:curricularUnit, aborda-se o conceito ***di:scientificArea***, que mapeia a noção de área científica na totalidade do modelo, conforme a Figura 3.20.

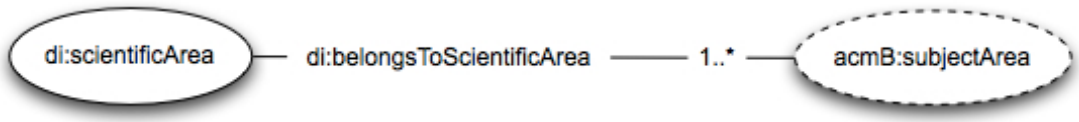


Figura 3.20 - O conceito di:scientificArea e relação com acmB:subjectArea

Naturalmente existem atributos simples, nome e descrição, para identificação da área, globalmente ao modelo. Como exemplo, refere-se "Informática", "Matemática", "Física", etc. Por fim, a relação ***di:belongsToScientificArea*** permite indicar quais dos conceitos *acmB:subjectArea* conhecidos pertencem à área científica em questão. Para as *subjectAreas* do CdC ACM-B, sabe-se que se estão na área científica de "Informática". Este conceito é também utilizado mais à frente, pelas estruturas curriculares.

O detalhe dos pormenores do conceito di:curricularUnit e respectivos conceitos associados, está descrito no documento anexo B.

3.3.4.2 O conceito *di:curricularUnitInstance*

Na sequência da descrição detalhada da UC, deverá ser apresentada a instância da UC, que tendo por base o modelo, descreve aspectos mais relacionados com a "execução" operacional efectiva da UC, nas suas várias vertentes anuais.

Como forma de representar o conceito da instância de UC, estabeleceu-se o objecto ***di:curricularUnitInstance***, que relembrando-se a Figura 3.13, estende *acmB:course* e *di:curricularUnit*, podendo assim reutilizar e redefinir os seus atributos e relações. Conforme se pode verificar na Figura 3.21, o conceito tem variados atributos e relações, que por uma questão de compreensão se optou por dispor na figura da seguinte forma: *i*) na zona superior direita da figura, os atributos da conceito, herdados da UC, que poderão ser modificados na instância – *di:cuAlias*, *di:Version* e *di:cuECTS*; *ii*) na zona esquerda superior da figura, as relações herdadas que se definiu "trancadas", e portanto não "modificáveis" na instância de UC (mas note-se, acessíveis pelo facto de existirem na definição de

UC) – *acmB:courseSyllabus* a *di:cuScientificArea*; *ii*) na zona esquerda central da figura, as relações herdadas de acmB:course, mas com possibilidade de alteração (dado serem definidas na UC) – *acmB:courseTitle* a *acmB:courseNotes*; *iii*) na zona direita central da figura, as relações herdadas de *di:curricularUnit*, que poderão servir para modificação dos dados definidos na UC – *di:cuAcmUnits*, *di:cuCourseSyllabus* e *di:gainedSkills*; *iv*) na zona inferior da figura, alinhadas, as relações novas da instância de UC – *di:cuStudentWork*, *di:cuCourseProgram*, *di:cuEvaluationModel*, *di:cuActivities* e *di:cuBibliography*; v) e por fim, imediatamente acima do conceito *di:curricularUnitInstance*, a relação *di:cuModel*, onde se especifica qual a UC utilizada para a definição desta instância;

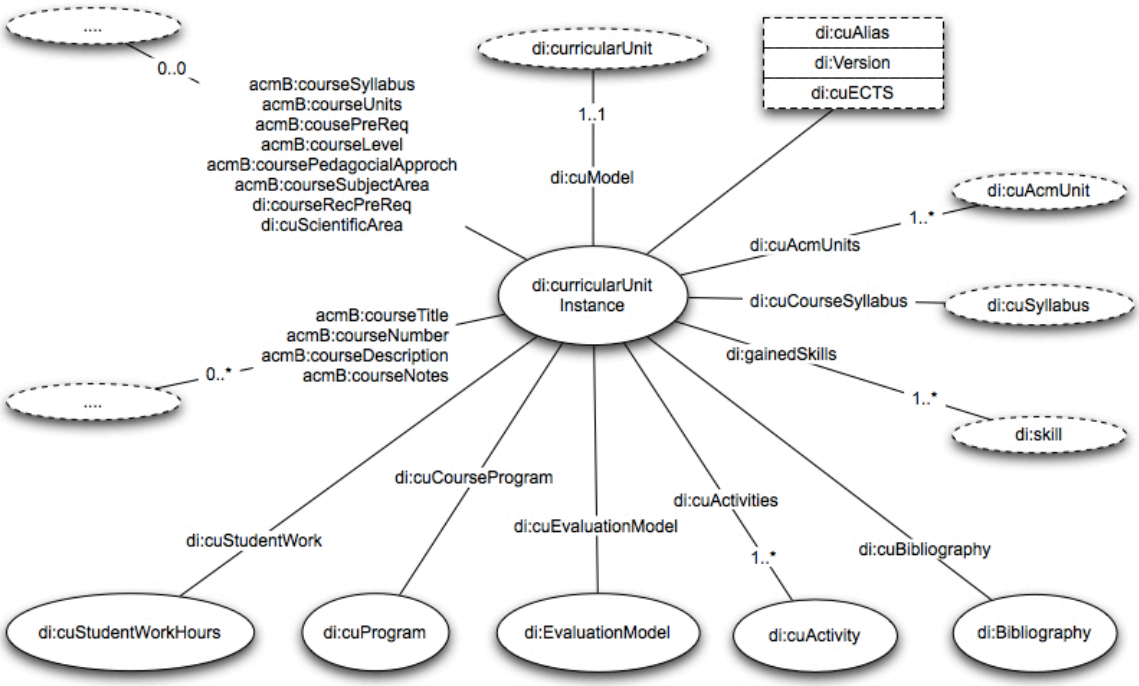


Figura 3.21 - o conceito *di:curricularUnitInstance* (instância de UC)

Antes de se iniciar a descrição das novas relações e conceitos específicos da instância de UC, descrevem-se apenas algumas considerações sobre as relações herdadas, considerando que a maioria mantêm a mesma semântica anterior:

- O atributo ***di:Version*** tem um significado similar ao atributo utilizado no modelo da UC, ou seja, uma indicação da versão da instância da UC, sendo igualmente um elemento do par "chave" das instâncias no modelo: <*acmB:courseTitle*, *di:Version*>. A proposta para o seu valor é considerar o ano da sua execução e outro valor numérico de controlo.
- As relações herdadas ***di:cuAcmUnits***, ***di:cuCourseSyllabus*** e ***di:gainedSkills***, permitem igualmente herdar a totalidade dos dados definidos na UC. No entanto, considerando as eventuais alterações que poderão surgir na operacionalização da instância, permite-se a

alteração destes dados já preenchidos com base na UC, garantindo que se cumpre a lógica originalmente definida para as relações.

Passa-se então à descrição das novas relações definidas no âmbito da instância de UC: *i*) ***di:cuModel***, que como o nome indica, permite ligar a instância da UC à respectiva UC que servirá como seu "template" e definição. Esta relação deverá ser estabelecida no momento de definição da instância, indicando assim qual será o modelo UC no qual a instância será baseada. Isto permite o preenchimento de informação de forma automática, nomeadamente as relações descritas atrás herdadas de di:curricularUnit (e logicamente, de acmB:course); *ii*) ***di:cuStudentWork,*** que estabelece uma relação da instância ao conceito *di:cuStudentWorkHours*, que detalhado mais à frente, permite que sejam definidos num único conceito o conjunto de atributos que refere o esforço do aluno na UC, medido em horas, com uma distribuição pelos diferentes tipos de actividades curriculares de uma disciplinas, sejam aulas práticas, aulas teóricas, estudo, avaliações, etc. (como habitual, os esforços em horas do aluno poderão ser convertido em ECTS, considerando o ratio horas / crédito); *iii*) Dado que no *syllabus* da UC, não é relevante a ordem, sequência ou qualquer tipo de agrupamento ou organização dos tópicos, isto não é suficiente para as considerações operacionais da instância, onde se pretende definir ordem e organização na sua abordagem. Para isso, definiu-se ***di:cuCourseProgram***, que estabelece uma relação ao conceito *di:cuProgram* (descrito em seguida), único em cada instância, permitindo efectivamente definir secções, agrupamentos e sequências desses mesmos tópicos; *iv*) ***di:cuBibliography***, que permite suportar a componente especialmente operacional da bibliografia na instância da UC. Estabelece-se uma ligação a um conceito *di:bibliography*, único em cada instância, onde se poderá indicar o conjunto de referências bibliográficas que serão declaradas pelo regente da disciplina como relevantes para a sua aprendizagem. Estas referências deverão ainda possibilitar a identificação de uma relação com as unidades da instância, estabelecendo assim um padrão unidade – referência, que poderá ser útil para utilizações noutras instâncias; *v*) ***di:cuActivities***, que permite definir outro aspecto existente na operacionalização das disciplinas, a identificação das diversas actividades e seus diferentes tipos que poderão ser desenvolvidos ao longo da execução da instância (aulas teóricas, aulas práticas, trabalhos de laboratório, avaliações via testes ou exames, etc.). Esta relação estabelece uma ao conceito *di:cuActivity*, que representa no modelo o conceito de actividade da disciplina e respectiva informação de detalhe da mesma; *vi*) ***di:cuEvaluationModel***, que permite definir outro aspecto da operacionalização das UC, o modelo de avaliação proposto, que tipicamente poderá variar de UC em UC, e que tem especial importância na forma como as *skills* são transmitidas aos alunos. Esta estabelece uma relação ao conceito, *di:evaluationModel*, que permite detalhar no modelo a forma avaliação proposta na instância.

Assim, tendo uma visão global da instância da UC, passar-se-á à descrição em detalhe dos vários subconceitos que a compõem, permitindo a recolha da informação adicional de operacionalização da UC.

Inicia-se pelo conceito ***di:cuStudentWorkHours***, que permite executar uma simplificação do modelo, fazendo o agrupamento de vários atributos relacionados com a quantificação aproximada (ou expectável) do esforço do aluno na UC, relativamente aos diferentes tipos de actividades de ensino que poderão existir dentro de uma disciplina. Veja-se a Figura 3.22, onde se apresenta o conceito e os respectivos atributos, que foram baseados numa estrutura anteriormente utilizada no DI. Esta está consubstanciada em fichas preenchidas para as várias disciplinas, tendo sido criada para a remodelação de Bolonha como uma primeira tentativa de modelação transversal do esforço às várias disciplinas DI, estando disponível na secção 11 do documento "Proposta de curso de 1o ciclo de Licenciatura em Engenharia Informática" [71].

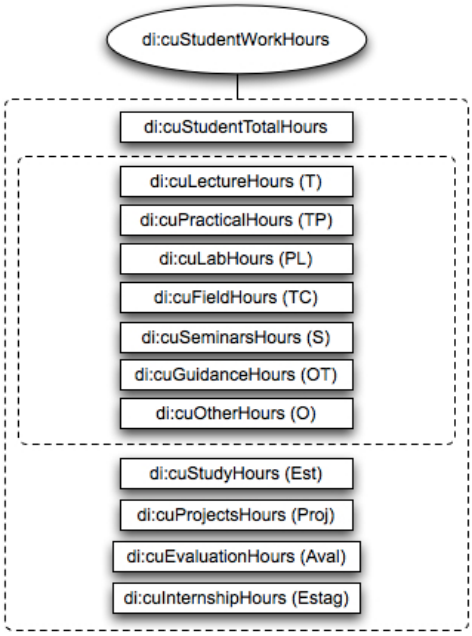


Figura 3.22 – O conceito *di:cuStudentWorkHours* (de esforço do aluno)

Apesar do esforço ter como medida de base ao longo do sistema a unidade de créditos (ou seja, o conceito ECTS), nestes atributos, por uma questão de simplificação da forma de especificação da instância, optou-se por utilizar a indicação do esforço como horas expectáveis de envolvência do aluno. Conforme já referido, estas poderão ser facilmente ser convertidas em ECTS. Por exemplo, se for considerando o valor horas / crédito utilizando na FCT-UNL, ou seja, 28 horas por crédito, uma UC que contabilize o esforço total do aluno ao longo das várias actividades em cerca de 222 horas, corresponde aproximadamente a 8 ECTS (*222 / 28 = 7,93*). É também importante voltar a enfatizar que as horas declaradas nestes atributos, tipicamente definidas pelo responsável da instância UC, são valores **expectáveis aproximados** do esforço que um aluno terá de desenvolver para cumprir com os objectivos da UC e ter uma avaliação positiva.

Cada um dos atributos poderá ser brevemente descrito como: *i*) ***di:cuStudentTotalHours***, o valor total do esforço do aluno na instância da UC. Deverá ser sempre o somatório dos diferentes esforços do aluno (os outros atributos), sendo este o único de preenchimento obrigatório; *ii*) ***di:cuLectureHours (T)***, o esforço do aluno em aulas consideradas como "teóricas". Geralmente calculado com base nas horas semanais das aulas multiplicado pelo número de semanas onde a disciplina será leccionada; *iii*) ***di:cuPraticalHours (TP)***, o esforço do aluno em aulas "teórico-práticas". Note-se que a existência deste tipo de aulas contra as aulas puramente "práticas", varia de acordo com a forma como a disciplina é abordada, em concreto, caso as aulas práticas tenham ou não também uma componente de abordagem teórica; *iv*) ***di:cuLabHours (PL)***, o esforço do aluno em aulas "práticas e laboratoriais". Estas aulas serão meramente práticas, ou seja, com abordagem prática à disciplina (onde não se abordará teoria) e tipicamente leccionadas em laboratórios específicos (a maioria das disciplinas do DI será deste tipo); *v*) ***di:cuFieldHours (TC)***, i esforço do aluno considerado como "trabalho de campo". Tipicamente um conceito mais utilizado em disciplinas das áreas sociais, refere-se a horas onde seja efectuado trabalho orientado pelo professor, mas fora do espaço habitual das salas de aulas. Por exemplo, uma visita de estudo ou um trabalho de inquérito directamente a pessoas na rua; *vi*) ***di:cuSeminarsHours (S)***, o esforço do aluno em "seminários" ou "palestras". Note-se que considerar algumas horas como seminários depende do regente da disciplina. Tipicamente horas que envolvam apresentações de trabalhos ou discussões sobre determinada temática podem ser consideradas aqui. Igualmente, podem também ser consideradas horas que envolvam discussões e apresentações com oradores convidados; *vii*) ***di:cuGuidanceHours (OT)***, o esforço do aluno em "orientação tutorial". Este esforço considera as horas onde se revê e discute com o professor os materiais apresentados nas aulas, bem como a preparação dos trabalhos a efectuar; *viii)* **di:cuOtherHours (O)**, o esforço do aluno em horas de outro tipo qualquer, não classificadas em nenhuma das categorias apresentadas; *ix*) ***di:cuStudyHours (Est)***, o esforço do aluno em horas envolvido no trabalho efectivo de estudo dos temas da disciplinas; *x*) ***di:cuProjectHours (Proj)***, o esforço do aluno em horas envolvidas para a execução prática de trabalhos e projectos da disciplina; *xi*) ***di:cuEvaluationHours (Aval)***, o esforço do aluno em horas de actividades de avaliação, entenda-se, exames, testes, provas orais, apresentações, etc., que tenham o objectivo de avaliar a aprendizagem das matérias transmitidas nas aulas; *xii*) ***di:cuInternshipHours (Estag)***, o esforço do aluno em horas de estágio. Aplicável tipicamente aos projectos finais (quando tenham essa componente) ou a cadeiras de estágio profissionalizante.

Por fim, para facilitar o preenchimento destes atributos, antevê-se uma interface que permita uma visualização automática entre ECTS e horas, e que garanta a lógica dos somatórios e ultrapassagem de limites.

Passando-se à descrição do conceito ***di:cuProgram***, relembra-se que este foi criado para permitir algo que o syllabus não faz, ou seja, uma organização por secções e sequenciação dos tópicos que serão abordados na UC, seguindo a ordem que o orientador pretender para a disciplina. Considerando a revisão do conceito syllabus, veja-se a Figura 3.23, onde se apresentam os vários subconceitos definidos.

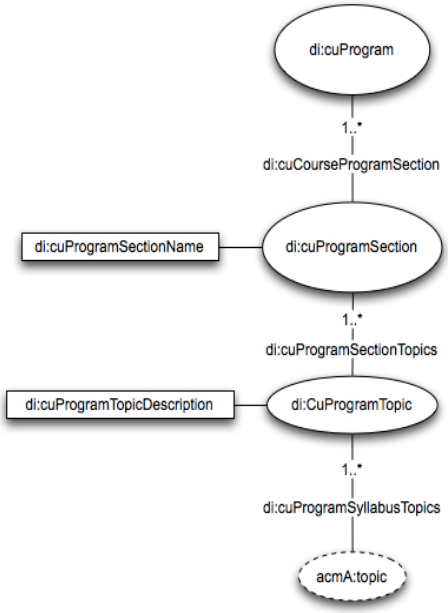


Figura 3.23 – O conceito *di:cuProgram* e respectivos sub-conceitos

Os vários subconceitos criados são relativamente lineares, conforme se pode verificar, iniciando-se pela existência de 1 ou mais secções dentro do *di:cuProgram*, que se relacionam pela ligação ***di:cuCourseProgramSection***. Logicamente, o conceito **di:cuProgramSection** especifica uma secção dentro da estrutura de um programa da instância da UC e permite ao especificador da UC (normalmente o regente da disciplina) agrupar os tópicos conforme as diferentes separações conceptuais que possam existir nas temáticas da disciplina. Não se estabelece nenhuma regra ou critério para a criação das secções, deixando estas totalmente da responsabilidade do especificador. Estas secções terão um nome, com recurso ao atributo ***di:cuProgramSectionName***, que as permite identificar univocamente dentro do programa da instância. As secções terão ainda ligações, com recurso à relação ***di:cuProgramSectionTopics***, onde se estabelecem quais os tópicos do programa que farão parte desta secção.

Estes tópicos do programa são definidos então pelo conceito ***di:cuProgramTopic***, que é uma evolução dos tópicos declarados no *syllabus* da UC, permitindo assim agrupar os vários tópicos ACM da UC, num objecto que retrata um tópico conceptual a abordar. Este conceito, em conjunto com a existência de secções, permitirá alguma flexibilidade na definição do programa da UC, garantindo assim um programa explícito e útil aos alunos. Estes tópicos do programa terão uma descrição, via o atributo ***di:cuProgramTopicDescription***, onde se permite uma descrição textual do tópico conceptual a abordar na

instância da UC. Por fim, existe uma relação ***di:cuProgramSyllabusTopics***, que permite ligar cada um destes tópicos conceptuais aos vários *topics* ACM declarados nos syllabus da UC que o constituem (ou seja, os *acmA:topic* indicados nas unidades a abordar). Obviamente, aqueles que tenham sido indicados dentro das unidades como tópicos (*acmA:topic*) a não serem abordados (via *di:notCoveredTopic*), não poderão ser incluídos no programa da instância da UC.

Na sequência dos subconceitos de *di:curricularUnitInstance*, apresenta-se o conceito ***di:Bibliography***, que suporta a componente bastante relevante do material de suporte à temática abordada na disciplina, seja em formato livro, apresentações, referencias *Web*, etc. A bibliografia de uma instância de UC deverá portanto ser entendida não apenas como uma listagem de livros, mas como um conjunto de referências bibliográficas de diversos tipos, que naturalmente abordarão cada um diversas das temáticas da disciplina.

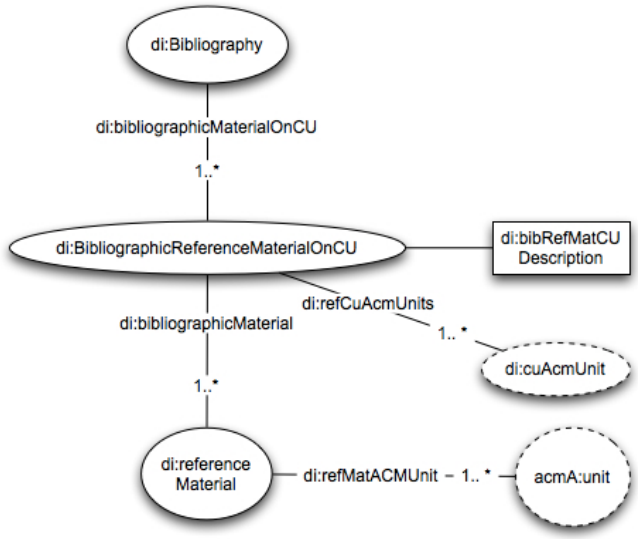


Figura 3.24 - O conceito di:Bibliography e seus subconceitos

Veja-se a Figura 3.24, onde se apresenta o conceito *di:Bibliography*, que representará a bibliografia da instância da UC, senda esta composta por uma lista de material bibliográfico, de acordo com a organização pretendida pelo responsável da UC, contendo também ligações às temáticas que estas referências abordam (via ligação a unidades ACM). Define-se assim uma relação ***di:bibliographicMaterialOnCU***, onde se referenciam vários conceitos **di:BibliographicReferenceMaterialOnCU**, que representam um material de referência na instância da UC. Este conceito não representa directamente uma referência, mas sim um elemento intermédio, com objectivos de organização e clarificação, no qual se permite relacionar vários tipos de referências bibliográficas (existentes globalmente no modelo), que podem abordar (ou não) o mesmo tópico ou tema. Com a criação desta forma de organização, pretende-se permitir alguma liberdade na especificação da bibliografia de acordo com a

preferência do regente da instância da UC. Este conceito tem um atributo, ***di:bibRefMatCUDescription***, onde é possível descrever genericamente (sobre a forma textual) o que se pretende para este material de referência nesta instância (por exemplo um agrupamento de determinado tema). Existem ainda duas relações neste conceito: *i*) Considerando que uma UC abordará diversas temáticas (representadas por unidades ACM ou *ad-hoc*) e que uma referência bibliográfica utilizada na instância, poderá abordar igualmente diversas temáticas (não sendo necessariamente as mesmas), fará sentido na indicar especificamente quais das temáticas da UC serão apoiadas pelo material de referência em questão. Para este efeito, define-se a relação ***di:refCuAcmUnits***, que no âmbito deste conceito, servirá para ligar este eventual "grupo" de referências bibliográficas (ou seja, o material de referência), às unidades ACM já especificadas na UC (normais ou *ad-hoc*), especificando assim quais delas as suas referências abordam e portanto a sua cobertura temática pretendida na instância (naturalmente, não serão ligados às unidades da UC, os temas das referências bibliográficas que não fazem fazer parte das temáticas da UC); *ii*) ***di:bibliographicMaterial***, a relação que permite especificar que referências bibliográficas fazem parte deste material de referência da instância, estabelecendo uma ligação a conceitos *di:referenceMaterial*.

Por fim, o conceito ***di:referenceMaterial*** representa globalmente no modelo (e portanto aplicável a qualquer conceito ou mesmo outros modelos) uma referência a material bibliográfico. Pretende-se apenas modelar suficiente e relevante neste conceito, pois considera-se a sua redefinição futura com objectivos de ter um formato reutilizável em diversas plataformas e sistemas. Por exemplo, refere-se o *Bibliographic Ontology Specification*[22] [96], o mesmo o standard *Dublin Core* [42]. O que se pretende no imediato para este conceito é ter a sua referenciação no modelo e, pelo facto de se saber que cada referência bibliográfica terá certamente correspondência nas unidades ACM representadas no corpo de conhecimento ACM-A, ter um mapeamento, mesmo que básico, desta relação (obviamente, não exclusivamente nem na totalidade). Para isso, define-se a relação ***di:refMatACMUnit***, que estabelece uma ligação entre o conceito de referência bibliográfico e as respectivas unidades ACM cobertas pela sua temática, via o conceito *acmA:unit*. Para detalhe da referência, e sabendo que se considera a sua futura remodelação, definiu-se apenas alguns atributos base auto-explicativos: *Title*, *Creator*, *Date*, *Description*, *Format, Publisher*.

Refere-se que este aspecto particular da relação da referência à sua temática poderá ser bastante interessante, pois poderá ser possível encontrar padrões de

[22] "*The Bibliographic Ontology describe bibliographic things on the semantic Web in RDF. [..] can be used as a citation ontology, as a document classification ontology, or simply as a way to describe any kind of document in RDF. [..] can be used as a common ground for converting other bibliographic data sources*".

utilização de referências bibliográficas tendo por base as suas temáticas, ou inversamente, recomendar referências tendo por base as temáticas da UC.

Segue-se então para o detalhe do conceito ***di:cuActivity***, que se representa na Figura 3.25.

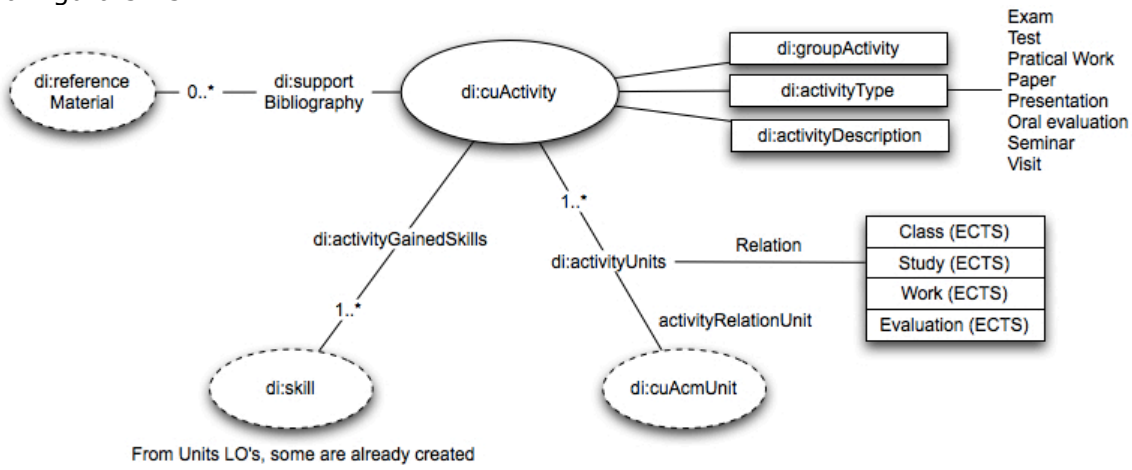


Figura 3.25 - O objecto *di:cuActivity* e suas ligações

A instância de uma UC, ou seja uma iteração da mesma em determinados anos / semestres lectivos, terá sempre associado um **conjunto de actividades mais práticas, extras às habituais aulas**, que serão desenvolvidas ao longo da execução da mesma (exemplo: testes, exames, relatórios, chamadas orais, apresentações, seminários, visitas de estudo, etc.). De forma a representar este conceito de "actividade numa UC", estabeleceu-se o objecto *di:cuActivity,* onde é recolhida informação de caracterização sobre a actividade (o seu tipo, descrição, se é uma actividade de grupo, cobertura temática, competências que se pretende transmitir ao aluno e qual o material de referência que suporta a sua execução ou aprendizagem).

Definem-se os seguintes atributos base: *i*) ***di:activityDescription***, uma descrição textual para maior detalhe da actividade em questão e do seu contexto na UC; *ii*) ***di:groupActivity***, que permite especificar a quantos elementos esta actividade se destina. O valor do atributo em actividades de aprendizagem é tipicamente de 1, pois estas são focadas unicamente ao aluno (apesar de todos os alunos que frequentam a UC serem os participantes alvos), enquanto que em actividades do tipo avaliação poderá ser indicado qual a dimensão do grupo alvo (exemplo: uma avaliação que consistem numa apresentação, em grupos de 2 elementos); *iii*) ***di:activityType***, que especifica os diversos tipos de actividade que poderão ocorrer: exames, testes, trabalhos práticos, relatórios, chamadas orais, apresentações, seminários, visitas de estudo.

Relativamente às relações do conceito, definiram-se as seguintes: *i*) ***di:supportBibliography***, onde é possível estabelecer uma relação para as referências bibliográficas (*di:referenceMaterial*, já associadas atrás à instância via o conceito *di:Bibliography*) que suportam a actividade a ser definida. Note-se que o facto de ser estabelecer uma relação a *di:referenceMaterial* e não ao conceito dessas referências na UC (*di:bibliographicReferenceMaterialOnCU)*, deve-se ao

facto deste último agrupar os materiais por temática ou por outro critério definido pelo especificador da UC, podendo cobrir mais, menos, ou não totalmente a temática da actividade. Note-se que, a indicação da bibliografia de suporte de uma actividade é de especial relevância, por exemplo, para efeitos de uma actividade de avaliação, esta indica ao aluno o que deve consultar para a aprendizagem da temática a ser avaliada; *ii*) ***di:activityUnits***, onde se pode indicar quais as unidades da UC estão envolvidas ou serão abordadas na execução de cada actividade em particular. Conforme se verifica na figura, esta relação tem algumas particularidades, algo que é descrito em seguida; *iii*) por fim, ***di:activityGainedSkills***, onde se indicam que competências (*di:skills*) da totalidade de competências da UC, terão origem na actividade a definir. Dado que já foram indicadas quais as unidades da UC são abordadas na actividade, é possível já indicar por inerência (e de forma automática) um conjunto de *skills*, na forma de *acmA:learningObjectives*, como sendo desenvolvidas pela actividade. Naturalmente, o especificador da UC poderá sempre adicionar novas *skills* e alterar esta afectação por inerência, sendo necessário garantir que estas estão também indicadas como *skills* globais da UC.

Falta então abordar a particularidade da relação *di:activityUnits*, que como se pode verificar na figura, consiste num elemento intermédio ***di:activityUnitRelation***, onde se definem 4 atributos, *Class*, *Study*, *Work*, *Evaluation*, algo que necessitam explicação. Da mesma forma que na indicação das unidades na UC se distribuiu o esforço do aluno (via ECTS) pelas unidades e, subsequentemente pelos 4 eixos em cada unidade, também para as actividades faz sentido efectuar o mesmo procedimento, permitindo assim identificar o tipo de esforço naquela actividade, para aquela unidade (temática) em particular. A semântica dos atributos mantêm-se a mesma, respectivamente, esforço relativo a aulas (no sentido genérico do conceito, teóricas, teórico-práticas, etc.), esforço do aluno relativo ao estudo (preparação para aulas, estudo para exames, etc.), esforço sobre os trabalhos desenvolvidos (trabalhos práticos ou entregáveis que não exames para avaliação), e esforço relativamente à execução da avaliação (testes, exames, chamadas orais, etc.).

Tendo estes atributos presentes, é relevante perceber que uma unidade da UC poderá estar presente em várias actividades. Assim, como exemplo, considerando uma unidade "i" anteriormente declarada na UC e o valor Xi, como o valor total de esforço do aluno na unidade, os respectivos eixos Ci,Si,Wi,Ei da unidade poderão ser distribuídos por qualquer actividade que inclua essa unidade "i". No entanto, considerando que as actividades declaradas são, como referido anteriormente, actividades mais práticas e não directamente "aulas", a distribuição do eixo *class* e *study* pelas várias actividades que abordam determinada unidade "i", poderá não corresponder à totalidade do valor Ci e Si da unidade. Por outro lado, e precisamente também porque as actividades retratam a totalidade das avaliações, a distribuição do eixo *work* e *evaluation* pelas várias actividades que incluem a unidade "i", deverá corresponder à totalidade do valor

Wi e Ei da unidade. Com base nisto, e na forma de distribuição de esforço pelos eixos da unidades, é possível estabelecer o conjunto final de regras para a distribuição do esforço da unidade ao longo das várias actividades, conforme se verifica na Figura 3.26.

$$ECTS^{UC} = \sum_{n} ECTS^{UC}_{unidade\_n}$$

$$ECTS_{unidade\_n} = Class^{ECTS}_{unidade\_n} + Work^{ECTS}_{unidade\_n} + Study^{ECTS}_{unidade\_n} + Evaluation^{ECTS}_{unidade\_n}$$

$$ECTS_{unidade\_n} \geq \sum_{i} ECTS^{unidade\_n}_{actividade\_i}$$

$$Class^{ECTS}_{unidade\_n} > \sum_{i} Class^{unidade\_n}_{actividade\_i}$$

$$Study^{ECTS}_{unidade\_n} \geq \sum_{i} Study^{unidade\_n}_{actividade\_i}$$

$$Work^{ECTS}_{unidade\_n} = \sum_{i} Work^{unidade\_n}_{actividade\_i}$$

$$Evaluation^{ECTS}_{unidade\_n} = \sum_{i} Evaluation^{unidade\_n}_{actividade\_i}$$

Figura 3.26 – Distribuição das ECTS pelas unidades e actividades da UC

Note-se o facto do esforço do eixo *Study* da unidade ser maior ou igual que o somatório desse esforço nas várias actividades. Isto deve-se ao facto das actividades serem maioritariamente práticas e de avaliação, sendo teoricamente possível que o aluno só necessite de esforço de estudo para as actividades indicadas e não concretamente para as aulas (mas não recomendável, obviamente). Estas regras definem um conjunto de indicações para a forma de introdução desta informação de esforço em cada unidade / actividade (que conforme referido anteriormente nas unidades, sejam via ECTS, via percentagens ou via horas), servirão para garantir a coerência global desta informação na UC e nas actividades.

Por fim, terminando os conceitos associados a *di:curricularUnitInstance*, descreve-se o conceito **di:EvaluationModel**, representado na Figura 3.27. Dado que nas actividades que são declaradas em cada instância de UC, conforme referido anteriormente, são também identificadas as actividades de avaliação dos alunos, é importante detalhar em pormenor o tipo e forma da avaliação da instância, algo que não é o objectivo na declaração de actividades. As formas e tipos de avaliação de disciplinas, poderão ter particularidades complexas, como por exemplo, frequências aos exames práticos que transitam de anos anteriores, despensas a exame, componentes práticas que influenciam a frequência nos exames teóricos, etc. Estes, e outros critérios ou regras de avaliação, definem o modelo de avaliação da instância, que tipicamente poderá variar entre UC, e que é de especial importância na forma como as *skills* são transmitidas aos alunos. Por este motivo, define-se o conceito *di:EvaluationModel*, onde é possível definir

para a instância qual o tipo de modelo de avaliação da mesma, seja este já pré-definido ou conhecido no modelo, ou definido manualmente pelo regente aquando da especificação da UC.

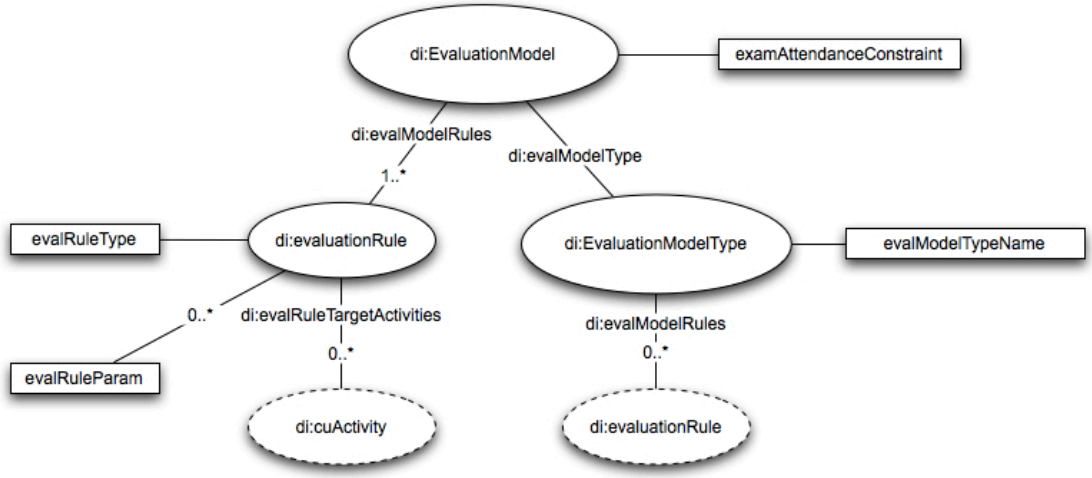


Figura 3.27 - O conceito di:EvaluationModel e seus subconceitos

Assume-se que a forma de avaliação de uma disciplina é geralmente constituída por um conjunto de **critérios** ou **regras** que deverão ser aplicadas (ou verificadas) sobre variadas actividades de avaliação que existem na UC. Estas actividades podem ser consideradas como o "alvo" das regras. Com base neste pressuposto, é possível definir também a ideia de tipos ou **modelos de avaliação comuns**, geralmente utilizados na universidade, que serão também constituídos por regras. Por exemplo: *i*) apenas 1 exame teórico, 100% da nota; *ii*) 1 exame teórico e um trabalho prático, 50% de distribuição da nota; *iii*) 1 Exame teórico e 2 trabalhos práticos e frequência, 50% para o exame e 25% para o trabalho, e frequência à cadeira nos trabalhos práticos (prática pode transitar a nota para anos seguintes); etc. Uma lista destes tipos de avaliações comuns poderá já existir pré-definida, facilitando o trabalho do regente especificador da instância.

Assim, é possível definir o conceito de forma de avaliação na instância da UC, *di:EvaluationModel*, o qual é constituído, na sua conceptualização básica, por regras de avaliação. Estas poderão ser definidas pela identificação de um modelo de avaliação pré-definido (que poderá já "alimentar" algumas das regras), ou pela indicação de um modelo "manual", onde o regente especificador da instância deverá indicar as regras manualmente.

Para isto, definiu-se no conceito um atributo ***di:examAttendanceConstraint***, que permite indicar se o modelo de avaliação da instância terá ou não a noção de frequência à disciplina (isto é, a possibilidade de o aluno não ir a exame, de acordo com a sua prestação em determinado conjunto de actividades de avaliação, por exemplo, trabalhos práticos, etc.). O conceito é ainda constituído por 2 relações: i) ***di:evalModelType***, onde se indica que a avaliação da instância tem por base um tipo de modelo de avaliação pré-definido, estabelecendo-se uma

ligação ao conceito *di:EvaluationModelType* (descrito mais à frente); *ii*) ***di:evalModelRules***, onde se estabelece ligações com conceitos *di:evaluationRule* (descrito igualmente mais à frente), permitindo assim definir as regras de avaliação da instância. Note-se que a indicação de um tipo de modelo de avaliação pré-definido (relação anterior), deverá automaticamente preencher as regras nesta relação. No entanto, caso algumas destas regras seja modificada manualmente pelo regente (entenda-se as suas definições base), o tipo do modelo de avaliação deverá ser alterado para "manual" (indicando assim que as regras já não são de acordo com o modelo pré-definido).

Descreve-se então o conceito ***di:evaluationModelType***, que sendo global a todo o modelo, permitirá definir vários tipos de modelos de avaliação pré-definidos, aplicáveis a qualquer instância de UC a definir. O conceito é constituído pelo atributo ***evalModelTypeName***, o qual define univocamente o nome do tipo de modelo de avaliação, e uma relação ***di:evalModelTypeRules***, onde se indicam as regras base do tipo de modelo de avaliação, com recurso a ligações ao conceito *di:evaluationRule*.

Na sequência descreve-se então o conceito ***di:evaluationRule***, que permite identificar as regras que fazem parte da forma de avaliação da instância (definido atrás). Este conceito, aplica-se genericamente a uma condição, validação ou restrição sobre determinada actividade de avaliação da instância, descrevendo uma das formas de avaliação que o aluno que frequenta a instância é submetido. Estas regras são uma declaração de intenção da **forma** como se pretende avaliar o aluno, preferencialmente explícita e precisa, para que possa ser comparável (exemplo: permitir cruzamento com outras instâncias e UC).

Define-se que as **regras são expressas de forma declarativa, tendo um tipo** (restrições, verificações, cálculos, etc.) **ao qual são aplicados parâmetros** (notas, frequências ou obrigatoriedade), **com uma actividade de avaliação "alvo"** (trabalhos, testes, exames, etc.).

Assim, o conceito *di:evaluationRule* é constituído por 2 atributos e 1 relação. Os atributos são: *i*) ***evalRuleType***, onde se identifica o tipo da regra; *ii*) ***evalRulesParam***, um conjunto de parâmetros que definem características da regra. A relação, ***di:ruleTargetActivities***, estabelece uma ligação às actividades "alvo" da regra, representadas pelo conceito *di:cuActivity* (naturalmente, actividades já definidas na instância).

Relativamente ao detalhe do tipo de regras e tipos de modelos de avaliação pré-definidos, refere-se para os pormenores do conceito di:curricularUnitInstance e respectivos conceitos associados, descrito no documento anexo C (digital).

3.3.5 Curso e Estruturas / Planos / Ofertas Curricular

No seguimento das secções anteriores, e como finalização do detalhe do modelo, descreve-se nesta secção os detalhes relativos aos aspectos de estruturação curricular. Considerando o referido anteriormente na secção 3.2, relativa à apresentação geral do modelo, as noções de estrutura curricular, plano

curricular, oferta curricular e curso, versão, perfil e ramo, serão agora consubstanciadas em conceitos com maior detalhe.

Inicia-se com a apresentação da Figura 3.28, onde se introduz a noção de *cursos*, um agrupamento de definições de curso, caracterizada sobre o conceito ***di:Programs***.

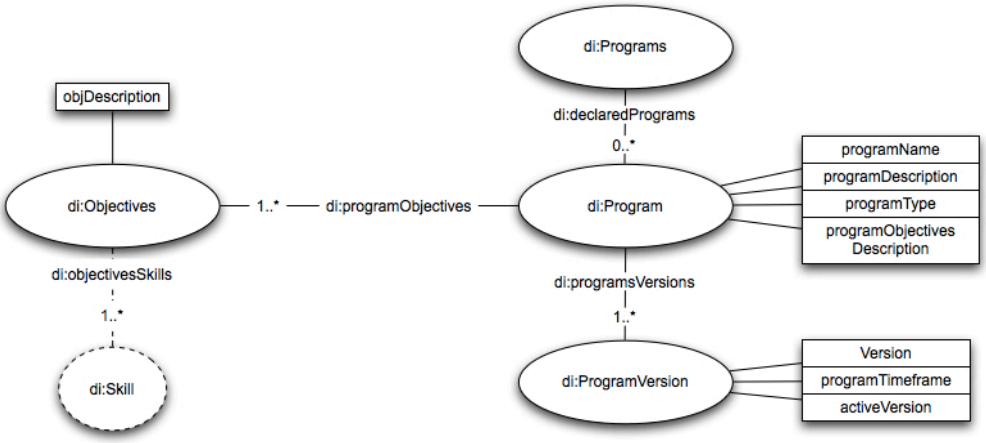


Figura 3.28 - Os conceitos *di:Programs*, *di:Program* e *di:ProgramVersion*

O conceito *di:Programs* é então responsável no modelo, pelo agrupamento de vários cursos, algo que na realidade ultrapassa os objectivos dessa tese (focada no problema da modelação de um curso 1º ciclo). No entanto, optou-se por considerar já esta possibilidade, eventualmente preparando o modelo para a definição eventual de cursos de 2º e 3º ciclo.

Existirá então uma relação ***di:declaredProgramms***, que estabelece uma ligação ao conceito *di:Program*, que efectivamente representa a noção de curso no modelo. Este conceito ***di:Program***, terá por sua vez um conjunto de atributos que permite definir o curso modelado: *i*) ***programName***, o nome ou designação do curso; *ii*) ***programDescription***, uma descrição do mesmo; *iii*) ***programType***, uma indicação do tipo de curso. Para o âmbito desta tese considerou-se que o tipo corresponderia à indicação do ciclo de Bolonha, 1º, 2º ou 3º; *iv*) e por fim, ***programObjectivesDescription***, uma descrição textual alto-nível dos macro-objectivos do curso (veja-se por exemplo a secção 4 do documento do DI, "Proposta de curso de 1o ciclo de Licenciatura em Engenharia Informática" [71]).

Como forma de tornar mais clara, e eventualmente comparável e verificável, a definição de objectivos proposta acima, estabeleceu-se a relação ***di:programObjectives***, onde através de ligações ao conceito *di:Objectives*, se poderá detalhar uma lista concreta dos objectivos propostos, que seguindo os conceitos de Bolonha, poderão inclusive serem mapeados com competências a atingir pelos alunos. É importante introduzir já este conceito ***di:Objectives***, que além do típico atributo ***objDescription*** onde é possível descrever o objectivo em detalhe, possui também uma relação para conceitos *di:Skill*. Esta relação, de nome ***di:objectiveSkills***, permitirá estabelecer o mapeamento de quais as competências que se pretende atingir com o objectivo definido. Note-se que estas competências, por ser definidas com recurso ao conceito *di:Skill*, poderão utilizar

os vários tipos e níveis de *skills* propostas no modelo. Este mapeamento permitirá ter definido um conjunto de competências guia para o curso, que poderão eventualmente ser verificáveis ou comparáveis entre cursos ou aspectos do curso (perfis, ramos).

Faltará apenas abordar a relação *di:ProgramVersions*, que através de uma ligação a conceitos *di:ProgramVersion*, permite a existência de versões do curso (noção já apresentada na secção 3.2), que se referem às diferentes iterações de especificação curricular que um curso poderá ter. O conceito ***di:ProgramVersion***, terá então 3 atributos que o descreve: *i*) ***Version***, uma descrição textual que especifica esta iteração de especificação curricular (exemplo: "Remodelação de Bolonha"); *ii*) ***programTimeframe***, onde se pode especificar qual o período ou prazo de execução previsto desta iteração; *iii*) e por fim, ***activeVersion***, que permite indicar se é a iteração/versão actualmente activa e a decorrer. Naturalmente, este conceito permite manter histórico das diferentes iterações de especificação curricular, e desde que seja garantido o mesmo cuidado nas especificações de UC e instâncias de UC, introduz-se no modelo a possibilidade da comparabilidade e melhoria contínua nos seus vários aspectos.

Tendo já sido introduzidas estas noções na visão geral do modelo, sabe-se que a versão de curso é efectivamente constituída e definida pela sua componentes de estruturação curricular. Em concreto, a estruturação de perfis e subsequentes ramos, que definem os aspectos estruturais do curso, e as estruturação das várias ofertas curriculares, que definem os aspectos das operacionalizações anuais do curso. Recorda-se a Figura 3.3 da visão geral e apresenta-se agora a Figura 3.29.

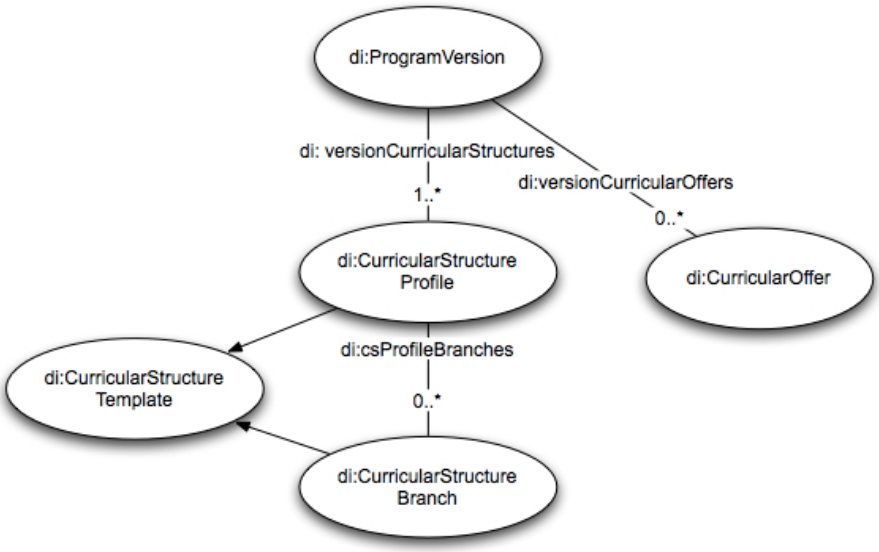


Figura 3.29 – O conceito di:ProgramVersion e seus subconceitos

O conceito *di:ProgramVersion* prevê então, além dos atributos acima mencionados, mais 2 relações: *i*) **di:versionCurricularStrutures**, onde se podem definir as várias estruturas curriculares da versão do curso, com recurso a

ligações ao conceito *di:CurricularStructureProfile*, que define a noção de perfis no curso (1 ou mais); *ii*) ***di:versionCurricularOffers***, onde se podem definir as diferentes ofertas curriculares anuais, ou seja, as implementações efectivas anuais da versão do curso definida. Note-se que poderão existir versões que ainda não tenham ofertas curriculares, por exemplo, em situações do decorrer da definição de uma nova versão, enquanto se efectuam experimentações e comparações com versões anteriores.

Mencionando o conceito ***di:CurricularStructureProfile***, que mapeia no modelo a noção de perfil, verifica-se que este define uma relação, ***di:csProfileBranches***, onde é possível indicar, caso existam, os ramos desse perfil. Esta relação estabelece ligações à noção de ramo, pela existência do mapeamento da mesma no conceito ***di:CurricularStructureBranch***. Terminando o apresentado na Figura 3.3 da visão geral, e agora na Figura 3.29, falta apenas descrever um aspecto relevante. A extensão, pelos perfis e ramos, do conceito *di:CurricularStructureTemplate*, que representa a noção de **Estrutura Curricular (EC)** no modelo, permite dizer que ambos serão eles próprios estruturas curriculares, com as possibilidades de definição e conceitos que dela se herdam.

Passar-se-á então à descrição do modelo relativo a estruturas curriculares (e respectivos planos curriculares), seguindo posteriormente da descrição do modelo relativo a ofertas curriculares.

3.3.5.1 Estrutura Curricular (EC) e Plano Curricular (PC)

A noção de estrutura curricular, já referida anteriormente, entende-se por um *template* que permite definir os aspectos estruturais de um *curriculum*. Com base nesta noção foi definido o conceito ***di:CurricularStructureTemplate***, que está representado na Figura 3.30 e o qual se passa a descrever e detalhar em seguida.

Este conceito, conforme já referido na visão geral do modelo, permitirá definir o *curriculum* mínimo obrigatório e os conjuntos/alternativas de disciplinas opcionais, tendo por base as UC definidas, bem como um Plano Curricular (PC) associado, que utilizará as UC indicadas neste conceito para estabelecer a organização recomendável desse *curriculum*, com a distribuição anual e semestral recomendada para as mesmas. Além destes aspectos, é necessário também considerar o suporte de outras estruturas que consolidam e melhoram as potencialidades do modelo, em concreto, a possibilidade de identificação de quais dos objectivos do curso se pretende para esta EC e a indicação das áreas científicas (e respectivos créditos mínimos necessários a cumprir pelo aluno), disponibilizadas nesta EC.

Ao conceito estão associados 3 atributos simples: *i*) ***csName***, onde se poderá atribuir um nome à estrutura curricular. Note-se que dada a existência da hierarquia de versões, perfis e ramos, uma eventual implementação deste modelo deverá lidar com a identificação e nomeação unívoca das EC; *ii*) ***csActive***, que

permite indicar se esta EC está "activa" ou não. Ou seja, se deve ser considerada como disponível para comparações, para a definição de ofertas curriculares com base nas suas UC indicadas, etc.; *iii*) e ***minECTS***, que permite indicar qual o valor de ECTS mínimos que os alunos deverão obter nesta EC.

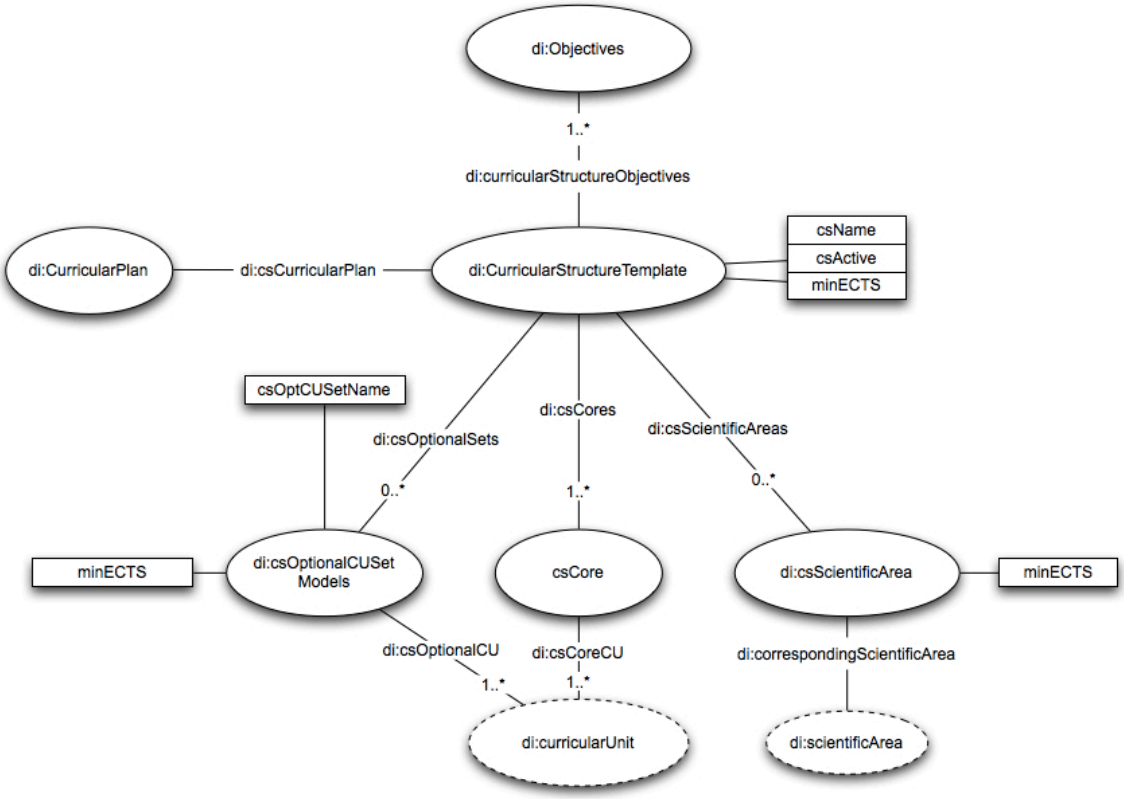


Figura 3.30 – O conceito *di:CurricularStructureTemplate* e seus subconceitos

Conforme a figura, são definidas também várias relações a partir do conceito: *i*) ***di:csCurricularPlan***, onde se estabelece uma relação ao conceito *di:CurricularPlan*, permitindo assim associar um plano curricular a esta EC; *ii*) ***di:curricularStructureObjectives***, onde estabelece uma relação para conceitos *di:Objectives* (referidos atrás), indicando quais dos objectivos previamente definidos pelo curso (*di:Program*) se prevê que sejam cumpridos por esta EC. Considerando a relação, já referida, dos objectivos com *skills*, está-se por inerência de indicar, mesmo que em mais alto nível, quais as *skills* que se pretende que os alunos adquirem nesta EC (algo que poderá ser teoricamente verificável pela obtenção de todas as *skills* das UC definidas nesta EC); *iii*) ***di:csScientificAreas***, onde através de uma relação para conceitos do tipo *di:csScientificArea*, é possível especificar, para as várias áreas científicas da EC (obtidas pelas suas unidades curricular definidas), se existem ECTS mínimos que os alunos deverão obter nessa mesma área para serem considerados como graduados na EC; *iv*) ***di:csCores***, onde através de uma relação para conceitos *di:csCore*, é possível indicar vários agrupamentos de unidades curriculares (UC) *core*, ou seja, definição do *curriculum* mínimo obrigatório para essa EC; *v*)

di:csOptionalSets, onde através de uma relação para conceitos *di:csOptionalCUSetModels*, é possível definir agrupamentos de unidades curriculares (UC) opcionais, ou seja, *curriculum* opcional.

Descrevem-se então em maior pormenor, os conceitos referidos imediatamente acima, iniciando-se pelo ***di:csScientificArea***. Com este conceito pretende-se definir a noção de área científica dentro da EC, permitindo retratar na EC eventuais restrições curriculares de ETCS mínimos impostos por área científica. Este terá um atributo, ***MANETAS***, no qual é possível especificar os ECTS mínimos impostos à área, e uma relação, ***di:correspondingScientificArea***, onde se estabelece uma relação com o conceito *di:scientificArea*, anteriormente descrito na definição da UC. Esta ligação permite indicar a que área científica das UC corresponde a área científica com restrições ECTS definida nesta EC.

Continuando, o conceito **di:csCore**, que define agrupamentos de unidades curriculares obrigatórias na EC, estabelece-os com recurso à relação ***di:csCoreCU***, que liga unidades curriculares (UC) previamente definidas.

Por fim, o conceito ***di:csOptionalCUSetModels***, que define agrupamentos de unidades curriculares opcionais, terá 2 atributos e 1 relação. Os atributos são, ***csOptCUSetName***, onde se define um nome para o conjunto opcional (com o âmbito restrito à UC), e ***minETCS***, onde se declara o valor mínimo de ECTS que o aluno deverá cumprir naquele agrupamento de UC opcionais. Naturalmente, a relação com o nome ***di:csOptionalCU***, estabelece ligação a conceitos UC, definindo assim as estruturas curriculares opcionais desse conjunto.

Faltam-nos apenas descrever um conceito da figura que ainda não foi referido, *di:CurricularPlan*. Este está descrito em seguida, tendo por base a descrição da Figura 3.31.

A definição de um plano curricular (PC), que existe sempre associado a uma estrutura curricular (EC), é feita com recurso ao conceito ***di:CurricularPlan***. Este consiste simplesmente na distribuição das várias UC, obrigatórias e opcionais, pelos vários anos e semestres previsto para a EC. Dado a representação da Figura 3.31 ser bastante explícita, apenas se refere que o plano é constituído por anos (*di:CurricularPlanYear*), e estes por semestres (*di:CurricularPlanSemester*). É possível em cada semestre indicar: *i*) via relação ***di:cpCoreCU***, a distribuição das UC obrigatórias já definidas na EC; *ii*) e via relação opcional ***di:cpOptCU***, quais os agrupamentos de UC opcionais, já igualmente definidas na EC, estão disponíveis.

Terminando esta subsecção dá-se um exemplo da utilização desta estrutura para modelar 2 cursos fictícios. Considere-se:

- 1 curso com 2 anos de "tronco comum" e o último ano com 2 "áreas" de investigação distintas: Isto poderia ser modelado com uma EC do tipo perfil definida para os 2 primeiros anos comuns e com 2 EC do tipo ramo, filhos do perfil, para o último ano.

- 1 curso com 2 perfis distintos, com regras de aprovação distintas, em que existem 2 anos de "tronco comum" e um último ano distinto: Isto poderia ser modelado com 2 EC do tipo perfil completamente distintas, ou 1 EC do tipo perfil para os 2 anos e 2 EC do tipo ramo para o último.

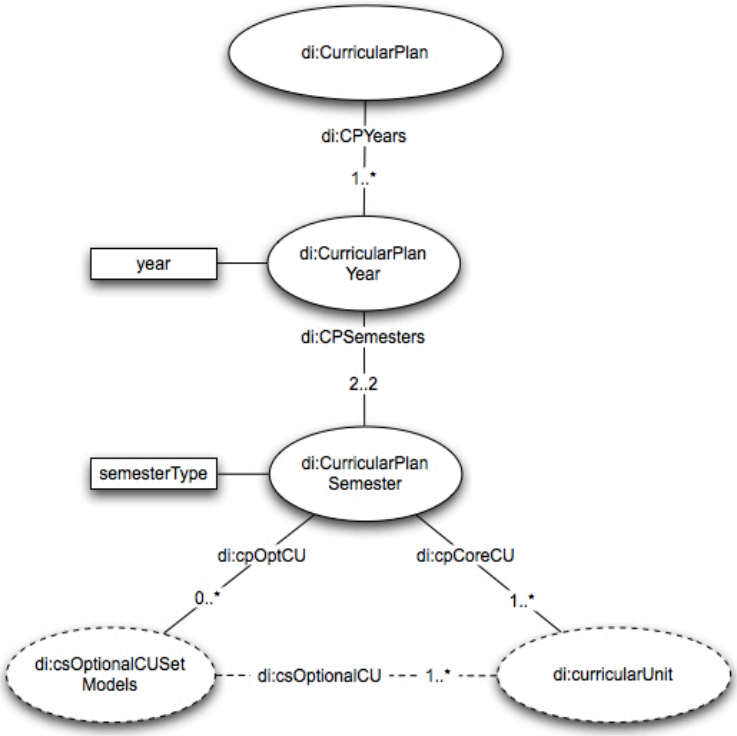


Figura 3.31 - o conceito *di:CurricularPlan* e seus subconceitos

3.3.5.2 Oferta curricular (OC)

Por fim, apresenta-se o conceito oferta curricular, que sendo conceito relacionado da versão do curso, corresponde às várias operacionalizações anuais dos cursos ou EC definidas, existindo em teoria uma definição por ano escolar.

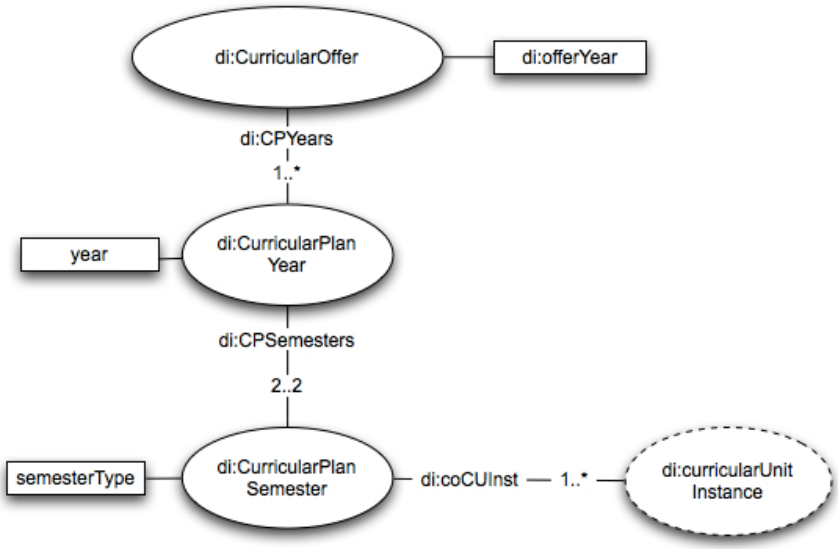


Figura 3.32 - o conceito *di:CurricularOffer* e subconceitos

A OC é mapeada pelo conceito ***di:CurricularOffer***, que é muito similar à noção do plano curricular (PC) descrito acima, conforme se pode verificar na Figura 3.32. Pretende-se que exista uma definição da OC por cada ano, onde se deverão associar as instâncias de UC disponíveis, ou leccionadas, nesse ano, tendo por base as respectivas UC indicadas nas EC activas definidas nessa versão do curso. Note-se que é apenas necessário indicar, quais as instâncias de UC a disponibilizar, em que anos e semestres, eliminando eventuais repetições geradas pela indicação da mesma UC (e subsequente instância de UC pretendida), nos diferentes perfis ou ramos eventualmente definidos.

Dado a lógica simples do conceito, apenas se referem o atributo ***di:offerYear***, que permite identificar univocamente a OC dentro da versão de curso (exemplo: "2009-2010"), e a relação *di:coCUInst*, que permite ligar aos conceitos di:curricularUnitInstance, estabelecendo a oferta com base nas UC das respectivas EC do curso.

Note-se que, uma eventual implementação deste modelo, devia considerar várias simplificações na fase de definição desta oferta curricular, disponibilizando automaticamente, por exemplo, uma lista de todas as instâncias das UC que existem declaradas nas EC activas, distribuídas por ano e semestre.

Terminando esta longa subsecção, indica-se que existe uma figura com a representação completa do modelo no documento anexo C (digital).

3.4 Povoamento do Modelo

Após o detalhe do modelo, nesta secção refere-se algumas considerações relativas ao povoamento do modelo com informação de diversas fontes. Este modelo proposto será tanto mais útil, quanto a qualidade e volume de informação nele disponível, pelo que para a sua melhor exploração, é importante considerar-se uma base mínima de informação pré-carregada.

3.4.1 Fontes de informação já identificadas

Tendo por base o apresentado até aqui, quer no Capítulo 2, os trabalhos relacionados, quer nas secções anteriores deste Capítulo 3, a descrição do modelo, é possível identificar um conjunto de fontes de informação que poderão "alimentar" partes do modelo proposto.

Considerando uma divisão pelas diferentes "áreas" do modelo, e respectivos conceitos relevantes, é possível identificar um conjunto de trabalhos que conterá fontes de informação para o seu povoamento (mesmo que parcial), conforme apresentação na Tabela 3.2.

Área do modelo	**Conceitos**	**Fontes**
Corpo de conhecimento ACM-A	*Area, Unir, Topic*	1. Trabalho ACM CC2001 – Apx A [29][30]
	Ligações entre *Area, Unit, Topic*	2. Trabalho "*Hyperkrep Academic Communities*" [62]
Corpo de conhecimento ACM-B	*Course*	3. Trabalho ACM CC2001 – Apx B [29][56]
Corpo de conhecimento *Skills*	LearningObjectives	1. Trabalho ACM CC2001 – Apx A [29][30]
	LearningObjectives e ligação a *Topics,* Verbos/Acções	4. Trabalho Taxonomia Bloom [74][75]
	Skills – DD's	5. Trabalho "Dublin Descriptors" [72]
	Skills – DD-e's e Skills genéricas	6. Trabalho "Criteria for Academic Bachelor's and Master's Curricula" [73]
	Skills – relação DD's e DD-e's	7. Trabalho "Proposta de curso de 1º ciclo de Licenciatura em Engenharia Informática" [71]
	Skills – soft e técnicas	7. Trabalho "Proposta de curso de 1º ciclo de Licenciatura em Engenharia Informática" [71]
Unidades Curriculares	(Parcial) UC e respectivas unidades temática. Instâncias de UC	7. Trabalho "Proposta de curso de 1º ciclo de Licenciatura em Engenharia Informática" [71]

Tabela 3.2 - Fontes de informação e áreas / conceitos de possível povoamento

Como forma de dimensionamento geral do esforço, apresenta-se para cada um dos trabalho identificados, um detalhe resumidos do eventual povoamento:

1. Trabalho ACM CC2001 – Apx A [29][30]
   Existe informação muito completa, sobre a forma de HTML e outras fontes computáveis (ou aproximadamente), dos conceitos ACM-A, pelo que se antevê um carregamento total da informação para os mesmos. Relativamente às *skills LearningObjectives*, poderá ser possível

efectuar algumas de ligações automáticas com *Topics*, sendo no entanto sempre necessário efectuar operações de validação manual.

2. Trabalho "Hyperkrep Academic Communities" [62]
   Os elementos deste trabalho, a pedido deste autor, disponibilizaram um conjunto de informação em formatos computáveis, que poderá servir para validar o trabalho ACM e introduzir as ligações entre os conceitos ACM-A, proposta originalmente por este trabalho.

3. Trabalho ACM CC2001 – Apx B [29][56]
   Existe informação muito completa, sobre a forma de HTML e outras fontes computáveis (ou aproximadamente), dos conceitos ACM-B, pelo que se antevê um carregamento total da informação para os mesmos, sendo porém necessário alguma revalidação manual das ligações.

4. Trabalho Taxonomia Bloom [74][75]
   A obtenção de uma lista de verbos da taxonomia de *Bloom* poderá axilar o preenchimento das ligações entre *LearningObjectives* e *Topics*, no que respeita à identificação das suas acções e classificações das *skills.*

5. Trabalho "Dublin Descriptors" [72]
   Permitirá um carregamento, já feito à partida pela subtipagem, das *skills DD's* no modelo.

6. Trabalho "Criteria for Academic Bachelor's and Master's Curricula" [73]
   Permitirá um carregamento, já feito à partida pela subtipagem, das *skills DD-e's* no modelo, a sua classificação dK, dS, dA e igualmente a introdução de *skills* genéricas com classificação DD-e.

7. Trabalho "Proposta de curso de 1º ciclo de Licenciatura em Engenharia Informática" [71]
   Este trabalho do DI, é uma fonte útil no povoamento de *skills*, seja no estabelecimento da relações entre DD's e DD-e's, ou na introdução de *skills* técnicas e *soft-skills* já identificadas no documento.
   Adicionalmente, a lista de disciplinas por este apresentada poderá servir de base para um carregamento manual de definições de UC e instâncias de UC, tendo obrigatoriamente de haver validações e entrega de informações adicionais pelos regentes para se estabelecer um povoamento completo da UC. Atreve-se no entanto a dizer que esta informação poderá servir de base inicial para o apoio a operações de definição extensiva.

3.4.2 Futuras fontes de informação a explorar

Relativamente a fontes de informação ainda não exploradas, identificam-se as seguintes:

- A necessidade de elaborar e desenvolver a componente da bibliografia de suporte às UC, carece de apoio de uma lista de referências bibliográficas focadas na área da ciência de computação. Eventualmente, aquando da definição das UC pelos vários dados disponíveis, e caso exista indicações de referências, antevê-se possível começar a explorar o povoamento desta componente.

- Com a revisão do CC2001 no CC2008 (ainda só em PDF) e as várias propostas de sub-*curriculums* (referidos no capítulo 2) podem, além de um corpo de conhecimento mais actualizado, introduzir um focado em várias subáreas. Estas, futuras, alterações são sem dúvida algo que deverá ser avaliado para revisão, quer dos dados da secção CdC (CC2008), quer mesmo da sua estrutura (sub-*curriculums*).

  O próprio conhecimento interno dentro do DI, relativamente aos detalhes das várias UC e do curso *ler-se*. Considera-se que a extensão das fichas do trabalho "Proposta de curso de 1º ciclo de Licenciatura em Engenharia Informática" [71], para recolha de mais informação de acordo com este modelo, poderia ser algo útil, caso existisse disponibilidade do corpo docente para o fazer. Com base nesta eventual informação "revista" das UC, seria possível efectuar uma modelação das várias estruturas curriculares com um bom nível de completude, deixando o modelo preparado para uma exploração operacional (entenda-se criação de instâncias e oferta curricular).

- Por fim, eventuais trabalhos de definição curricular, ou sistemas similares, desenvolvidos noutras universidades (Portuguesas e mesmo Europeias), poderiam ser fontes a investigar. Estes consideram-se relevantes, quer para aspectos de validação do modelo, quer para melhoria da informação do corpo de conhecimento (tópicos, *skills*, etc.).

3.5 Exploração do Modelo

Esta última secção do Capítulo 3, fecha com um enunciar de diferentes formas de utilização e exploração do modelo proposto. Pretende-se aqui apresentar as diferentes possibilidades do modelo, e de certa forma validar a premissa da estipulada no Capítulo 1, que a definição de um modelo computável, "*com a respectiva definição e normalização de conceitos do ensino, aliado à recolha de*

informação sobre a mesma base, permite produzir instrumentos úteis para a especificação, apoio à especificação e de análise sobre a completude, cobertura e qualidade dos cursos do ensino universitário".

Além dos usos imediatamente óbvios no apoio à definição curricular, onde existe a possibilidade de construção de interfaces que façam uso das relações existentes entre as diversas peças do modelo (exemplos: *units* e *topics* identificados noutras UCs, material de referência relacionado, apoio das pré-existentes definições *acmB:course*, obtenção de *skills* tendo por base as *units* da UC, etc.), é possível referir outras várias explorações possíveis de um modelo povoado com informação.

Optou-se por dividir esta secção em duas partes distintas: A 1ª, onde se descrevem explorações e utilizações possíveis sobre o modelo proposto, com base na informação de povoamento já conhecida (secção 3.4.1); E a 2ª, onde se considera a existência de uma implementação do modelo, integrada numa arquitectura com várias camadas e capacidades de ligações a outros sistemas para consumo e publicação de dados, com interfaces experimentadas e apoiado por um sistema de produção relatórios, gráficos, etc.

3.5.1 Utilizações "imediatas" possíveis

Assume-se nesta subsecção, que temos uma implementação do modelo descrito nas secções anteriores, povoado com as fontes de informação já conhecidas e identificadas (secção 3.4.1), e incluindo: *i*) O preenchimento de toda as informações da estrutura UC proposta, para todas as disciplinas actualmente em vigor no DI; *ii*) Uma definição completa da estruturação curricular do curso, versão, perfis ou ramos, que reflictam a situação actual no DI; *iii*) e uma definição da oferta curricular de, por exemplo, 2 últimos anos, com as respectivas instâncias de UC (relacionadas com as UC acima), e com a sua informação preenchida.

Este cenário, com a implementação do modelo e respectivo povoamento indicado, desde que se obtenha a colaboração dos regentes do DI para o ponto *ii*, é algo perfeitamente possível de atingir em curto espaço de tempo, daí considerar-se uma utilização "imediata". Assim, tendo por base as assumpções acima descritas, podem-se identificar **algumas** possíveis explorações "imediatas" do modelo proposto:

- **Cobertura temática de UCs** – é possível obter detalhes dos temas a abordar nas várias UC, seja via *units* ou *topics*, sendo por isso possível fazer comparações entre as várias UC definidas no curso;
- **Mapa temático detalhado do curso** – igualmente, devido às relações das UC, é possível obter um mapa muito detalhado dos tópicos (e unidades ACM) abordados no curso;
- **Verificação de cruzamentos de temas abordados nas UC do curso** – tendo este detalhe, é possível verificar onde existem cruzamentos ou "*overlapping*" dos temas nas UC do curso, no sentido

de perceber se podem ser feitas optimizações, reutilização de recursos, etc.;

- **Cobertura de competências das UCs** – da mesma forma que com os temas, também com base nas actuais relações da UC, é possível fazer comparações sobre competências, sejam comparações entre UC, ou mesmo um mapa de competências do curso;
- **Identificação de *skills* expectáveis nos alunos** – igualmente como no ponto acima, tendo por base as competências, é possível efectuar distribuições das mesmas nos alunos que tenham frequentado determinadas disciplinas, ou que tenham frequentado, ou estejam a frequentar, determinado ano/semestre, etc.;
- **Validação dos objectivos do curso** – dado a possibilidade de definição de objectivos do curso e relação dos mesmos com *skills* a obter, é possível quer apoiar a definição de UC do curso (para cumprimento dos mesmos), quer o inverso, validação dos objectivos com base nas competências totais definidas;
- **Apoio às escolhas curriculares dos alunos** – dado as relações de temas e competências nas UC, e o mapa "provável" das mesmas no percurso dos alunos entre N disciplinas, é possível apoiar os alunos na escolha de disciplinas opcionais, dando a possibilidade de escolher quais as *skills* mais desejáveis ou temas mais interessantes a explorar para maximizar o seu percurso académico;
- **Distribuições de ECTS nas UC** – é possível obter mapas de ECTS entenda-se esforço (em horas ou ECTS), quer ao nível da sua distribuição global em todas as UC do curso ou em todos os temas a abordar no curso (dada a relação de ECTS nas *units* das UC). É possível também efectuar distribuições pelos tipos de ECTS (*di:StudentWork*) e pelos tipos de actividades extra das UC (*di:cuActivity*). Naturalmente, isto permite fazer comparações entre UC, mapas de distribuição no curso, nos anos / semestres do curso, por actividade ou tipo de esforço, etc.;
- **Apoio à ligação de referências bibliográficas** – sabendo que as referências têm ligações com temas (*units*), é possível apoiar a definição de referências em novas UC no sentido de reaproveitar conhecimento anterior. Aliás, considerando a existência possível no futuro de uma lista de referências com ligações acmA:unit (ou algo similar), é ainda possível propor referências ao regente que define a UC com base nos seu temas;
- **Apoio à definição temática de novas UC (ou revisões)** – tendo um corpo sólido definido de UC, instâncias de UC, uma base de *acmB:course*, um curso definido, 2 anos de oferta curricular, etc., é possível obter uma base interessante para apoio de novas definições ou revisões de UC. Com isto poderão ser efectuados em "tempo real"

mapas comparativos da definição em execução com outras UC, verificar a forma com esta nova UC afecta a cobertura de temas, competências, ECTS do curso, do ano, do semestre, etc.;

- **Comparações das iterações do curso** – com base na oferta curricular e instâncias de UC, é possível comparar os diferentes anos lectivos entre si ou entre iterações (ou seja, 1º vs. 2º ano, ou anterior 1º no vs. actual 1º ano, etc.). Isto permitirá chegar às diferenças ao nível da cobertura de temas (*units* e *topics*), competências, esforço, etc.;
- **Detalhes do percursos expectáveis dos alunos** – igualmente relacionado com o ponto anterior de apoio às escolhas dos alunos, é possível com base nas estruturas dos cursos e ofertas curriculares, estabelecer o percursos expectáveis dos alunos, indicando quais os temas cobertos e competências esperadas nos diversos anos e semestres, e sua diferenças. Imagine-se por exemplo um mapa de evolução comparativo, onde é possível definir um curso, ramos ou perfis, ou mesmo 2 tipos de alunos distintos (ex: diferentes escolhas de UC opcionais), organizados por anos/semestres, onde se mostra as diferenças de temas e competências adquiridas;
- **Comparações de cursos que utilizem a base ACM** - tendo por base a informação ACM (areas, units, topics, learning objectives), é possível comparar o curso (perfis, ramos, etc.) com outros cursos (portugueses ou não), que utilizem a mesma base;
- **Comparações de cursos com base nos ECTS** – sabendo que os ECTS é uma medida europeia, considera-se que poderá ser utilizado para comparações com os cursos das várias universidades que adoptaram Bolonha. Sendo estes associados às UC não só pela sua atribuição, mas também distribuição nas suas unidades temáticas e actividades (e via atributos *Class, Study, Work, Oval*), é possível estabelecer diversas comparações. Além das expectáveis comparações ECTS dos anos / semestres, caso existem mais detalhes no(s) outros(s) curso(s), é possível comparar, por exemplo, diferenças as próprias abordagens pedagógicas (enfoque no estudo vs. enfoque em trabalho, etc.), esforço nas diversas áreas temáticas, etc.;
- **Utilizações com alunos ERASMUS** – existindo esta base de ECTS na Europa, a situação dos alunos que desejam frequentar o programa ERASMUS [97] poderá ser apoiada pela utilização de um modelo deste tipo, permitindo detalhe nas escolhas das UC (relativo por exemplo aos temas a abordar, detalhes no esforço envolvido, competências pretendidas, etc.). Igualmente para os serviços académicos, este tipo de modelos poderá ser útil ao apoio dos alunos ERASMUS, caindo em situações do ponto anterior (i.e., comparar o curso de origem dos

alunos com o *curriculum* actual, via comparação da distribuição de ECTS, dos entre temas, das competências, etc.);

- **Apoio às transferências de alunos** – existindo esta capacidade de identificar claramente os temas e competências adquiridos por determinados percursos dos alunos, é possível apoiar os departamentos das universidades (foca-se naturalmente o DI) nos processos de transferências de alunos, quer dentro da universidade, quer entre universidades;
- **Apoio à normalização de CdC CS a nível nacional** – Considera-se que o actual CdC proposto neste modelo, poderá ser eventualmente uma base relevante para esforços nacionais de normalização de temas e competências entre as universidades que disponibilizam cursos de CS. Os benefícios deste tipo de normalização são claros, podendo inclusive apoiar eventuais esforços a nível europeu a serem desenvolvidos por entidades com a associação *Informatics Europe*[23] [101].

Sugeriram-se então algumas das explorações possíveis, sem necessidade de revisão/extensão do modelo ou grande envolvimento (seja esforço ou tempo) na obtenção de dados para povoamento do mesmo, mostrando a sua utilidade e ao mesmo tempo tentando responder à premissa identificada no início desta secção.

3.5.2 Explorações possíveis do modelo num futuro sistema

Além do possível actualmente com o modelo proposto e povoamento com base nas fontes já identificadas, pode-se considerar algo mais ambicioso, assumindo a existência de uma implementação deste modelo numa arquitectura bastante mais evoluída.

Assume-se um sistema com as seguintes componentes: *i*) Um sistema *Web*, cliente/servidor de 3 camadas [98]; *ii*) Interface de interacção com o sistema, com tempo de experimentação "real" e várias iterações de refinamento; *iii*) Multi-utilizador, com controlo de acesso por perfis às diferentes áreas e capacidades do sistema; *iv*) Uma componente para produção de relatórios e gráficos sobre a informação e conteúdo do sistema, também com a possibilidade de geração de relatórios a pedido (além de pré-definidos); *v*) Uma componente para acesso da informação do modelo por sistemas externos, ou seja, uma *API*[24] que permitam outros sistemas "consumirem" informação deste (ex: *Web Services* [99]); *vi*) Uma componente para acesso a outros sistemas externos, com objectivo de povoamento de dados/conceitos no modelo (sejam novos ou revisões).

[23] "*Informatics Europe is the association of computer science departments of universities and research laboratories, public and private, in Europe and neighbouring areas"* [101]

[24] *API - Application Programming Interface*

Consideram-se dois tipos de acessos, a capacidade de execução de *API*'s externas e um "motor" optimizado para busca, consumo, processamento de dados externos (ex: *Web Crawler* [89]); *vii*) Um "motor" com algoritmos para procura de relações ainda não estabelecidas dentro do conteúdo do modelo (exemplo: motor de inferência [102], algoritmos de classificação automática de texto como *Naive Bayes classifier* [103] ou *Latent semantic indexing* [104], etc.).

Tendo estas características, considera-se que um sistema deste tipo poderá introduzir novas formas de exploração do modelo além das anteriormente referidas. Por exemplo:

- **Acesso à informação do CdC por sistemas externos** – a existência de uma *API* de acesso ao modelo permite disponibilizar a base de conhecimento dos temas (ACM-A e ACM-B) e das *skills*, a outros sistemas, outras universidades, outros departamentos da universidade, etc., tornando-se um apoio no esforço de normalização e utilização desta informação de uma forma mais abrangente;
- **Consumo e povoamento de informação bibliográfica** – A possibilidade de se estabelecer ligações a sistemas externos, inclusive considerando um cenário de "busca" periódica, poderia ser útil para o povoamento da informação de referências bibliográficas mais actualizadas e adequadas aos temas a abordar. Esta actualização considera-se útil aos especificadores e regentes das UC;
- **Actualização assistida do repositório CdC** – considerado as capacidades de busca e processamento de dados externos, bem como o "motor de inferência", pode-se estipular que é possível automatizar parcialmente o processo de povoamento de novas actualizações à informação do CdC, sejam temas, competências, etc.
- **A apresentação e visualização do modelo de forma evoluída –** a existência de uma componente de relatórios com camada gráfica e possibilidade de geração de relatórios específicos a pedido do utilizador, bem como pré-definidos no sistema, introduz um complemento muito interessante às várias explorações do modelo atrás declaradas. Considera-se por exemplo a possibilidade de relatórios que além de formatos textuais ou tabulares, poderão produzir gráficos de barras, tarte, radiais, etc., com a capacidade interactiva de navegação na informação, noções de navegação entre hierarquias (ex: *drill up* ou *down* [100]), etc.
- **Costumização do acesso ao modelo pelos perfis de utilizador** – a noção de multi-utilizador permitiria introduzir diferentes perfis de acesso ao sistema, alguns mais focados na especificação de UC ou EC, outros na manutenção do CdC, outros apenas no acesso ao conteúdo do modelo, sendo possível a costumização de relatórios e interfaces especificamente orientadas à função;

- **Integração com sistemas operacionais da universidade** – a capacidade de acesso a outros sistemas, ou de outros sistemas a este, introduz a possibilidade de se estabelecer ligações entre as explorações de informação dos percursos dos alunos, sua competências e temas abordados, com a real execução dos alunos, registada em sistemas operacionais dos vários departamentos académicos. Poderá ser assim possível nos sistemas operacionais detalhar, além das disciplinas frequentadas e notas, quais as competências adquiridas e temas abordados de determinado aluno específico, bem como por exemplo, a integração de relatórios comparativos no momento de escolha de novas disciplinas num semestre, etc.;
- **O "motor de inferência" e o seu impacto no sistema** – a existência de um componente capaz de procurar relações ainda não definidas no sistema pode ser muito útil para apoio aos processos de definição de UC, EC, instâncias de UC, etc. (ex: imagine-se no preenchimento de um *syllabus* o sistema analisar o texto e recomendar *topics* ACM relacionados, imagine-se que a indicação de material de referência poderá ser feita também pela análise das unidades UC e dos sumários ou outros textos indicativos da referência, etc.). Além disto, o próprio povoamento do CdC poderá beneficiar disto, pois no momento de introdução de novas informações CdC, o sistema poderá analisar e propor relações descobertas);
- **Melhorado apoio às transferências de alunos e ERASMUS** – sabendo que existem capacidade de ligações a outros sistemas e um motor de descoberta de relações com o conteúdo do modelo, pode-se assumir que tendo uma lista de disciplinas originais de um aluno "transferido" (e alguma definição base das mesmas), é possível identificar temas, tópicos e competências relacionadas com a informação do modelo, chegando por isso a uma base de trabalho inicial (que carecerá naturalmente de ser validada de forma manual);
- **Melhorado apoio à normalização de CdC CS (nacional/europeu)** – estas características de acesso à informação do sistema, bem como a descoberta "automatizada" da nova informação do mesmo (relações e busca), poderá apoiar fortemente a consolidação e normalização de um CdC da área CS, quer a nível nacional, quer a nível europeu.

Capítulo 4
Arquitectura proposta e Protótipo

Apresenta-se uma proposta de uma arquitectura "ideal" para suportar o modelo detalhado na secção anterior, bem como detalhe da construção de um protótipo exemplificativo de algumas das características e explorações possíveis do modelo e da arquitectura proposta.

Este capítulo terá dois objectivos específicos, a apresentação de uma proposta de arquitectura que suporte o modelo apresentado na secção anterior e o detalhe da construção de um protótipo exemplificativo, que demonstra algumas da características do modelo e algumas das suas possíveis explorações.

Consideram-se 4 secções, onde se abordará na primeira o âmbito e requisitos da arquitectura, a segunda onde se descreverá em maior detalhe a arquitectura proposta, a terceira onde se apresentam algumas tecnologias utilizadas no protótipo e, por fim, a quarta onde se descreve o protótipo implementado, as várias acções de construção e algumas das explorações obtidas.

4.1 Âmbito e Requisitos

Conforme referido atrás, o modelo proposto terá ainda mais utilidade se for suportado por uma arquitectura que permita melhorar o apoio à definição curricular e as explorações possíveis do mesmo.

Pretende-se que esta arquitectura proposta seja uma base realista, de implementação possível, que suporte as principais operações ao modelo na sua forma actual, sem cair necessariamente em características mais complexas como as descritas na anterior secção 3.5.2. Considera-se que a arquitectura estará integrada no âmbito de um Departamento de Informática de uma universidade que disponibilize cursos CS (o DI é assumido como exemplo), efectuando uma integração do modelo no apoio às principais operações de definição curricular. Por fim, a arquitectura pretende ainda contextualizar algumas noções para o protótipo, que demonstrará apenas algumas das características da arquitectura e algumas explorações do modelo, o suficiente para se demonstrar a viabilidade do modelo e da prova de conceito.

Descrevem-se então de forma geral, e em muito alto-nível, os seguintes requisitos base para uma arquitectura que, estando integrada no âmbito descrito e que suportando o modelo, permita as operações base referidas ao longo do documento até aqui:

- **Estabelecer um formato para a estruturação das temáticas de conhecimento e competências na área de CS**, ou seja, implementação do modelo e sua componente CdC;
- **Normalização do corpo de conhecimento** de forma que seja possível a **sua eventual partilha com outros sistemas,** departamentos ou estabelecimentos de ensino. Ou seja, um formato bem definido e documentado, com uma componente de partilha desses dados;
- **Capacidade de extensão do corpo de conhecimento** interna ao DI, de forma a lidar com a evolução natural da área. Ou seja, necessário suporte tecnológico adequado às capacidades de extensibilidade do modelo e versionamento adequado;

- **Capacidade de povoamento do corpo de conhecimento** com base em **informação externa**. Ou seja, necessário suporte tecnológico que utilizando as características de extensibilidade do modelo permita a alimentação do mesmo com base em fontes externas (considera-se uma intervenção manual, e portanto operações semi-automatizadas).
- **Estabelecer um formato para recolha da informação das disciplinas do curso**, que seja compatível (aproximadamente) com o que já existe no DI, **e que permita apoiar definição e revisão da definição curricular das mesmas e de suas iterações anuais**. Ou seja, implementação do modelo e suas componentes UC e instância de UC;
- **Estabelecer um formato para recolha da informação de estruturação de um curso**, que seja compatível (aproximadamente) com o que já existe no DI, **e que permita apoiar a definição e revisão da definição do mesmo e das suas iterações anuais**. Ou seja, implementação do modelo e das suas componentes EC/PC (curso, versão, perfil, ramo) e OC;
- Preferencialmente **um sistema *Web***, ou seja, uma arquitectura cliente/servidor de 3 camadas [98], **com suporte multi-utilizador e controlo de acesso por perfis às diferentes áreas e capacidades** da arquitectura;
- **Apoiar o mais possível os processos de definição curricular**, ou seja, além do recurso a informação normalizada, **garantir que as interfaces são desenhadas de forma a apoiar as escolhas efectuadas na definição, com acesso facilitado às várias peças e técnicas de ajuda automatizada** (cálculos automáticos, relacionamentos recomendados, escolhas pré-propostas, etc.);
- **Garantir** também os requisitos anteriores de **extensibilidade aos formatos de informação das disciplinas e dos cursos**, tendo em especial atenção os aspectos relativos a histórico, versionamento e activação / desactivação das mesmas.
- **Garantir** também os requisitos anteriores **de normalização e partilha aos formatos de informação das disciplinas e dos cursos**, aspecto que deverá ser integrado na componente de partilha de dados;
- **Capacidade de produção e execução de relatórios** sobre a informação do sistema, em formato tabular ou gráfico, com a possibilidade da adição de relatórios futuros;
- **Mecanismos básicos para procura automatizada de relações ainda não estabelecidas** dentro da informação recolhida. Naturalmente as relações encontradas necessitarão de validação manual. Pretende-se com isto melhorar a qualidade da informação do modelo.

4.2 Arquitectura base proposta

Tendo esta definição muito geral dos requisitos propostos, é possível iniciar o detalhe da arquitectura proposta e das várias componentes que a constituem.

4.2.1 *Overview* da Arquitectura

Tendo presente os requisitos anteriormente descritos, é possível esboçar uma descrição alto nível dos principais componentes propostos na arquitectura, sem entrar em aspectos relacionados com detalhes tecnológicos, conforme a Figura 4.1.

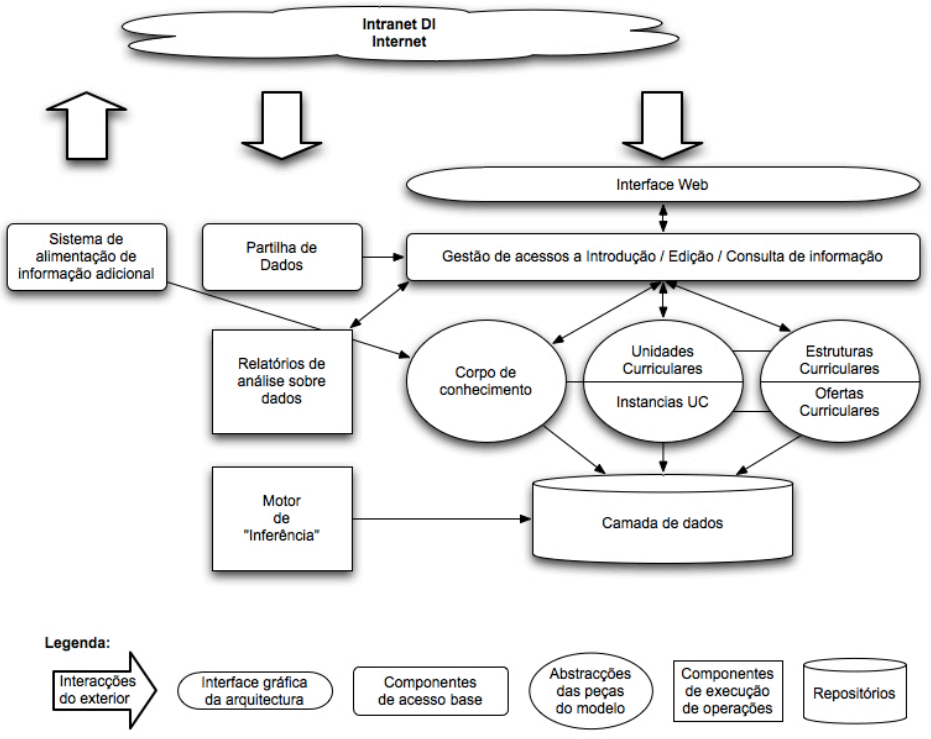


Figura 4.1 – Representação esquemática da arquitectura proposta

Faz-se então uma descrição breve da figura, detalhando-se posteriormente cada componente, indicado que requisitos cumpre e quais as suas principais funções, seguindo a lógica apresentada na legenda.

As setas principais representam interacções do exterior com o sistema, ou seja, acessos dos utilizadores ou de outros sistemas, ou mesmo deste sistema a outras fontes. **A interface gráfica da arquitectura** cumpre o requisito de acesso *Web* ao sistema por utilizadores, devendo ser uma componente onde se investirá fortemente em formas de apoiar a definição curricular e acessos simplificados à informação.

As componentes de acesso base ao sistema permitem disponibilizar as operações principais do sistema, devendo estas ser invocadas por qualquer interface ou mecanismo de interacção externo: *i*) **Gestão de acessos à Introdução / Edição / Consulta de informação**, que disponibiliza as

operações básicas de acesso ao sistema pelos utilizadores (e respectivamente a interface gráfica). Esta componente, além da disponibilização das operações principais, deverá garantir toda a lógicas de acesso multi-utilizador, permissões e perfis, etc., cumprindo assim este requisito; *ii*) A componente de **Partilha de Dados** disponibiliza as operações de partilha da informação com o exterior, devendo interagir com as operações principais do sistema (o ponto anterior) e efectuar quaisquer operações de transformações necessárias para respeitar formatos de partilha. Esta componente cumpre assim o requisito de partilha de informação do modelo com sistemas externos; *iii*) A componente do **sistema de alimentação de informação adicional** existe para lidar com a evolução natural dos temas na áreas, disponibilizando formas de actualização preferencialmente automáticas ou assistidas por validação humana, para povoamento do CdC com nova informação, cumprindo assim este requisito. Esta componente deverá disponibilizar mecanismos de ligação a fontes externas, bem como operações de transformação e limpeza da informação obtida para povoamento em informação útil no CdC. Considera-se que as operações poderão ser invocadas a pedido ou periodicamente, podendo posteriormente invocar operações na componente motor de "inferência".

As **componentes de abstracção do modelo**, representadas por círculos, respectivamente CdC, UC / instancias, e EC / OC, disponibilizam todas as operações base necessárias para lidar com estes elementos do modelo. Isto cria abstracções operacionais dos mesmos, para acesso pelas outras componentes da arquitectura, o que simplifica a construção da arquitectura. Estas componentes contém aquilo que se pode chamar de "lógica de negócio" do modelo, devendo lidar com aspectos de construção, extensibilidade, histórico, versionamento e activação, cumprindo assim estes risquitos.

As componentes de execução de operações, deverão ser entendidas como componentes para execução de operações muito específicas, que deverão ser invocadas pelas outras componentes, sem comunicação directa com o exterior: *i*) A componente para **relatórios de análise** é aquilo que permite a execução e armazenamento de relatórios sobre os dados do modelo, sejam em formato textual, tabular ou gráfico, cumprindo assim o requisito enunciado. O ideal será ter esta componente invocada pela camada de operações do sistema, que deverá fornecer os dados via abstracções do modelo, recebendo um relatório já produzido; ii) A componente do **motor de "inferência"** (usa a expressão de forma mais livre, sem conotação tecnológica), é responsável por operações de procura de novas relações ou co-relações dentro do modelo e da sua informação, utilizando diversos tipos de algoritmos. Assume-se que estas possam ser executadas periodicamente ou a pedido, produzindo um conjunto de informação que necessitará ou não de validação manual, de acordo com os algoritmos e as opções dos responsáveis do sistema. O requisito pretende apoiar a descoberta de mais informação pelos especificadores curriculares e utilizadores do sistema.

Por fim, a **Camada de Dados** considera-se como um repositório da totalidade de dados, que deverá lidar com os aspectos de armazenamento de informação e, se possível (dependendo da sua tecnologia), lidar com alguns dos aspectos extensibilidade, versionamento e evolução, apoiando assim as componentes acima.

Refere-se o anexo B (digital), que contém algum detalhe adicional sobre processos da arquitecta.

4.2.2 Detalhe da arquitectura e seus componentes

Após um *overview* da arquitectura, é possível entrar-se em algum detalhe acerca da forma como se propõe que a mesma seja construída, detalhando já alguns aspectos técnicos dos vários componentes.

Considerando os requisitos já enunciados anteriormente, a solução deverá ser baseada numa infra-estrutura de aplicação *Web*, considerando tecnologias que permitam alguma flexibilidade na construção dos componentes bem como na evolução do repositório de informação (de acordo com o requisito da extensibilidade). Não se pretendendo definir já uma implementação associada a qualquer tecnologia específica, apresenta-se aqui uma representação orientada ao desenvolvimento de uma aplicação Web, com o típico modelo cliente / servidor *three-tier* [98], orientando a disponibilização de operações com o exterior via *WebServices* [99], bem como a construção das componentes da forma mais modular possível.

De forma a se consubstanciar uma efectiva solução para suportar a arquitectura proposta na Figura 4.1, é possível detalhar cada um dos elementos representados originalmente de forma abstracta, em componentes concretos numa arquitectura *Web* 3 camadas, conforme se representa na Figura 4.2.

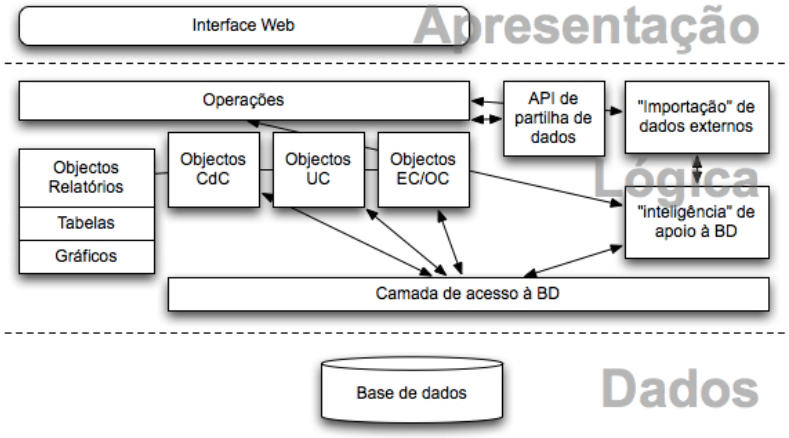


Figura 4.2 – Arquitectura proposta numa solução *Web* 3 camadas

A arquitectura é então constituída pelas 3 camadas típicas: *i*) **a camada de apresentação**, onde se considera as várias *interfaces Web* que permitem a integração dos utilizadores com o sistema, recorrendo às funcionalidades do nível abaixo; *ii*) a **camada lógica ou aplicacional**, onde estão agrupadas as

componentes da aplicação executam as várias operações do sistema; *iii*) a **camada de dados**, responsável pelo armazenamento correcto da informação da arquitectura, que deverá também ter, se possível, características particulares para lidar com a evolução e extensão do modelo.

Passa-se então a descrever os aspectos mais técnico e funcionais das várias componentes que existem na camada aplicacional, essencialmente onde residirá a "inteligência" do sistema, descrevendo-os em sequência *top-down*.

A componente "operações", transversal à maior parte do sistema, permite a comunicação com a camada gráfica e deverá disponibilizar todas as suas operações de acesso os elementos principais e secundários do modelo (CdC, UC, EC/PC, etc.), sejam de criação, acesso, adição ou remoção. Entenda-se esta componente como um conjunto de "serviços" que poderão ser invocados pelos outros componentes, executando as várias operações e garantido são controlados os aspectos de controlo de acessos, multiutilizadores, etc. Esta camada deverá também disponibilizar "serviços" de administração do sistema, como os aspectos de autenticação, a gestão de utilizadores e acessos/perfis, definição de relatórios, configuração da disponibilização de dados ao exterior, etc. Considere-se que esta componente poderá não ser necessitaria mente implementada *WebServices*, pois simplificará a camada de apresentação que tem a responsabilidade de invocar estas operações e "gerar" as páginas respectivas.

A componente *API* de partilha de dados deverá disponibilizar uma *API* para operações de consulta à informação por sistemas externos, preferencialmente implementada como *Web Services* [99]. Os *Web Services* deverão invocar as operações da componente anterior, ou seja, "espelhar" as operações de consulta e disponibilização de informação, efectuando quaisquer outros passos adicionais necessários. Esta componente deverá garantir a autenticação / identificação dos sistemas que a interrogam (recorrendo, se necessário à componente "operações"), e efectuar quaisquer transformações de formato necessárias (disponibilizar a informação em formatos XML [63], JSON [105], etc.).

Na componente de **"importação" de dados externos**, é possível podem definir e configurar fontes externas para obtenção de dados para povoamento do CdC (inicialmente, sugere-se apenas o foco no CdC), como ficheiros, invocação de URL ou API remotas. A cada fonte de dados deverá esta associado um conjunto de operações de busca, transformação e limpeza que resultará na invocação, automática ou com validação manual, de operações de criação e ligação de nova informação, quer pela componente "operações", quer pelos objectos base, quer por acesso directo à camada BD. Assume-se ainda que estes processos de povoamento poderão utilizar a componente do motor de "inferência" para melhorar a informação obtida. Esta componente poderá ter diferentes formas e complexidade, recomendando-se inicialmente uma versão simplista que poderá fazer uso de soluções já existentes de transformação de informação [108], mas

podendo evoluir também para *Web Crawlers* [109] ou soluções de *Text Mining* [110] mais evoluídas.

Considerando os vários tipos de conceitos que existem no modelo, e as questões de extensibilidade a ter presente, faz todo o sentido criar uma camada de abstracção que seja independente do formato e das especificidades dos mesmos. Definiu-se assim **as componentes dos objectos do modelo**. Entenda-se estes objectos como a representação lógica dos conceitos do modelo na arquitectura, garantindo o seu correcto preenchimento (atributos, relações e subconceitos) e funcionamento, focando-se nas operações executadas sobre estes. Estes objectos disponibilizam operações base CRUD[25] sobre cada conceito e quaisquer outras operações necessárias sobre os sub-conceitos e relações, mapeando a sua execução correcta ao modelo. É também da responsabilidade destes objectos lidar com as questões de versionamento, histórico e activação/desactivação dos conceitos, algo que poderá ser delegado à componente de acesso à BD (descrita à frente). No entanto é importante estes objectos serem desenhados já com considerações para a extensibilidade futura dos conceitos (ex: em vez de invocar relações e atributos pelo nome, lidar com as listas de ambos, a recepção de parâmetros como listas ou vectores, etc.).

Relativamente à **componente dos relatórios**, optou-se por especificar também uma abstracção de objectos que os representam na arquitectura, onde são disponibilizadas diversas operações para a sua definição e execução. Um relatório definido no sistema poderá ser executado por qualquer utilizador, sendo invocado tipicamente pela componente "operações", devendo disponibilizar o seu resultado (independente do formato) de uma forma previamente conhecida, para utilização pela interface ou pela API de partilha de informação (respeitando naturalmente as limitações destas). Considera-se a existência de sub-componentes de relatórios, responsáveis pela geração de diferentes tipos de resultados ou formatos (tabulares, gráficos, etc.), e pela sua disponibilização ao restante sistema (ex: *interface*). Isto será possível estender no futuro, por exemplo, considerar gráficos interactivos. Define-se que os relatórios são criados pelos responsáveis do sistema, indicando-se quais os dados a obter da camada de objectos do modelo ou da camada de acesso à BD, quais operações necessárias à agregação ou conversão dos dados, e qual o formato final resultante (invocando a sub-componente respectiva).

A componente de **"inteligência" de apoio à BD** é algo que, podendo ser executado pelas outras componentes da arquitectura, é responsável por identificar novas relações (ou informação) dentro dos conceitos e dados do modelo. Propõe-se inclusive, para o futuro, que qualquer alteração de dados seja analisada para a indicação de relacionamentos ainda não identificados, permitindo melhorar a qualidade do modelo. Esta componente deverá disponibilizar operações para analisar a informação de um conceito ou sub-conceito (existentes

[25] *CRUD - Create, Read, Update and Delete* [106] [107]

ou a introduzir), retornando possíveis relações com outros dados. Sugere-se disponibilizar este tipo de operações por tipo de conceito (ex: aos *learningObjectives*), ou por secção (ex: a qualquer elemento no CdC). Conforme se pode perceber, estas operações poderão ser executadas, de forma automática, por invocação, ou a pedido de um utilizador responsável, produzindo um conjunto de informação a qual poderá ser avaliada manualmente ou automaticamente introduzida no sistema. Este último aspecto variará naturalmente de acordo com os algoritmos e as opções dos responsáveis do sistema. Esta componente poderá fazer recurso de diversos tipos de tecnologias, como motores de inferência [102] na camada aplicacional ou na camada de dados, várias noções de *texto mining* [110], classificação de texto recorrendo à aprendizagem automatizada [111], ou mais recomendável para uma abordagem inicial, algoritmos de classificação automática de texto como *Naive Bayes classifier* [103] ou *Latent semantic indexing* [104].

Por fim, a componente ou **camada de acesso à BD**, deve ser entendida como uma abstracção do acesso à base de dados ou repositório de dados, simplificando as operações necessárias sobre estes. Considera-se que nesta arquitectura esta componente terá duas funções relevantes: *i*) se possível apoiar as componentes dos objectos em aspectos de versionamento e histórico (ex: manter sempre as versões anteriores a quando de alterações, etc.); *ii*) disponibilizar uma abstracção do funcionamento base do modelo (entenda-se, a noção de conceito, atributos, relações entre conceitos, etc.), utilizada pelas outras componentes da arquitectura. Apesar de se definir a arquitectura independente do tipo de base de dados a utilizar na camada abaixo, considera-se que a utilização de tecnologias semânticas poderá simplificar esta componente, dado que o modelo segue aproximadamente esta lógica (igualmente referido na secção seguinte).

Para terminar desta secção, pode-se então considerar que foi apresentada uma proposta de arquitectura recomendada para suporte do modelo, que tenta cumprir realisticamente com as explorações do mesmo propostas na secção 3.5. Considera-se que se fez uma descrição funcional das componentes base, tendo em atenção a não associação a tecnologias específicas.

4.3 Tecnologias associadas

Tendo então apresentado a arquitectura nas secções anteriores, descrevem-se agora muito resumidamente algumas tecnologias que serão utilizadas pelo protótipo para a implementação de algumas das explorações e componentes da arquitectura. As tecnologias são descritas apenas de forma introdutória pois não é este o objectivo da tese, e sim a prova do conceito, algo que as tecnologias apenas suportam.

4.3.1 RDF

O RDF, Resource Description Framework [112], é um *standard* W3C que permite a descrição de recursos com base em declarações. Em RDF, uma declaração, também referida por triplo ou triple to, é composta por 3 partes: Um recurso, uma propriedade e um valor (da propriedade). Utiliza-se também regularmente a nomenclatura sujeito, predicado e objecto.

Os recursos identificam objectos (no sentido conceptual) com recurso a *Uniform Resource Identifier* (URI), as propriedades descrevem atributos do recurso identificado pelo URI, e por fim, o valor corresponde ao valor atribuído à propriedade, que pode ser um valor literal ou um "apontador" para outro recurso, utilizando URI. Um documento RDF é então um conjunto de declarações RDF, podendo existir diversos formatos ou sintaxes para a sua definição, como RDF/XML [114] ou Notation3 (N3) [115].

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://en.wikipedia.org/wiki/Tony_Benn">
    <dc:title>Tony Benn</dc:title>
    <dc:publisher>Wikipedia</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

Figura 4.3 - Exemplo da sintaxe RDF/XML [116]

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.

<http://en.wikipedia.org/wiki/Tony_Benn>
  dc:title "Tony Benn";
  dc:publisher "Wikipedia".
```

Figura 4.4 - Exemplo da sintaxe N3 [116]

Apresentam-se exemplos desta notações na Figura 4.3 e Figura 4.4, sendo ambas utilizadas nos trabalhos do protótipo e referidas mais à frente. Refere-se também que, dada a construção por triplos, é possível apresentar estas declarações ou documentos como grafos de relações entre recursos, contendo atributos com valores literais, algo que de imediato estabelece um paralelo à lógica de construção do modelo.

4.3.2 RDF-Schema

O RDF Schema, ou RDFS [117], é igualmente uma recomendação W3C de uma linguagem utilizada para a construção ou definição de vocabulários RDF. Esta permite a definição de classes, subclasses e propriedades, de forma similar à noção de Objectos das linguagens de programação, permitindo a definição de hierarquias de classes, algo que também estabelece um paralelo à construção do modelo.

4.3.3 OWL

A *Web Ontology Language*, ou OWL [66], já referido anteriormente, é uma família de linguagens para a definição de ontologias, que foi desenhada para a utilização por aplicações que necessitem de processar informação, promovendo a interoperabilidade *Web* entre sistemas. Esta é descrita por formatos como RDF e

RDFS (e suas sintaxes possíveis), com a adição de conjunto de vocabulários, já tendo sido referida anteriormente no trabalho "*Hyperkrep academic communities*" [61] para a modelação do conhecimento CC2001 como uma ontologia.

A família OWL é composta por 3 sub-linguagens: OWL *Lite*, OWL DL e OWL *Full* [66]. As diferenças residem essencialmente nas capacidades de expressividade de alguns aspectos como cardinalidade, igualdade, etc., sendo apenas importante referir que o OWL Full não garante uma computação finita, pelo que se for necessário considerar noções de inferência, é recomendada a utilização de OWL DL.

Dada a sua utilização no trabalho acima referido, esta é também de especial importância no protótipo, pois cumpre com as necessidades de construção do modelo proposto.

4.3.4 O editor Protege

O Protégé [118] é uma ferramenta open source para a edição de ontologias e uma *Framework* para a modelação de conhecimento, desenvolvida pela *Stanford Center for Biomedical Informatics Research* da *Stanford University School of Medicine*. Este permite a modelação de ontologias de várias formas, suportando diversos formatos e sintaxes como RDF, RDFS e OWL. Uma das suas componentes, Protégé-OWL, está especificamente orientado à criação de ontologias em OWL, permitindo a definição de classes, propriedades, etc., bem como a utilização de motores de inferência (externos ou não) para descoberta de novo conhecimento em informação (instâncias) modelada pela ontologia. Apresenta-se um exemplo da *interface* da versão 3.4 na Figura 4.5, onde estão declaradas algumas classes OWL.

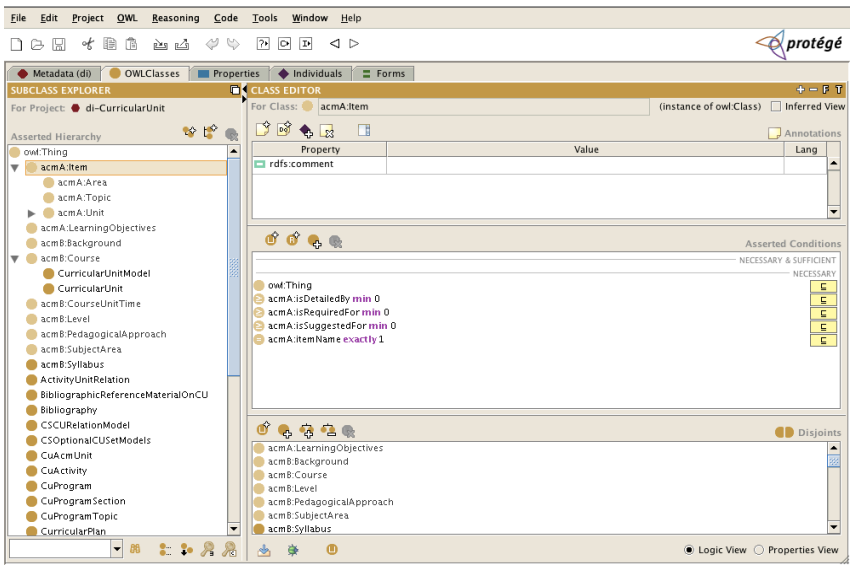


Figura 4.5 - Exemplo da Interface Protégé-OWL, versão 3.4.

4.3.5 Repositórios RDF nativos (*triplestore*)

Com a utilização de RDF surgiu a necessidade de investigar brevemente quais as melhores formas de armazenamento desta informação, podendo existir suportada sobre base de dados relacionais [119] [120] ou suportada como base de dados especificamente orientadas para armazenamento e obtenção de RDF, referidos como repositórios RDF nativos ou *triplestores* [121]. Estes permite acesso à informação com base em linguagens de *query* (ex: SPARQL [122]), mas estão especificamente optimizados para o armazenamento de triplos na forma referida anteriormente. Fizeram-se algumas pesquisas sobre diversas implementações, referido especificamente os seguintes: *Jena* [123], *Redland* [124], *Sesame* [125].

4.3.6 Tecnologia específica para o protótipo

Além das tecnologias referidas anteriormente, que se focam mais especificamente nos aspectos de modelação e armazenamento, e portanto mais aplicadas à construção do modelo, faz sentido referir algumas tecnologias utilizadas especificamente no protótipo. Em concreto refere-se a tecnologia utilizada para a construção do protótipo como uma aplicação *Web*, a *Framework* RubyOnRails [126], e a biblioteca *ActiveRDF* [131] para lidar com RDF.

4.3.6.1 Ruby e *RubyOnRails*

Para construção do protótipo, que se pretendia ser apenas uma explicitação curta que desse suporte à prova de conceito do modelo, optou-se pela utilização de uma *Framework open-source* para o desenvolvimento de aplicações *Web*, focada na produtividade e obtenção muito rápida de resultados, o *RubyOnRails* [126] (também conhecido apenas por *Raios*).

Esta *Framework* foi criada em 2003 por *David Heinemeier Hansson* [127], tendo por base o seu trabalho na empresa 37signals [128], e é construída sobre a linguagem *Ruby* [129]. Esta *Framework* utiliza o padrão de arquitectura *Model-View-Controller* (MVC) como base na construção de aplicações. Especificamente no Raios, *Models* são objectos abstracções de conceitos na camada de dados (BD), *Controllers* são aquilo que responde tipicamente a pedidos URL do utilizador (gerados por pedidos via interface ou invocação directa) e que executa operações, e *Views* são as interfaces, tipicamente respostas dos *controllers* devolvidas ao browser. O *Rails* introduz também vários apoios às tarefas de desenvolvimento, como a noção de *scaffolding* (construção automática de modelos e interface com base na definição e informação dos repositórios de dados), servidores aplicacionais *Web* integrados (como *WEBrick* e *Mongrel*), um sistema de apoio à construção (*Raie*) e à actualização e descoberta de componentes e *plugins* (*Gems*). O *Rails* faz uso intensivo de bibliotecas de *Javascript* para apoio à construção das *interfaces* e orienta-se a uma construção de Web Services RESTOU [130].

4.3.6.2 ActiveRDF

O *ActiveRDF* [131] [132] é uma biblioteca desenvolvida na linguagem *Ruby*, quer permite aceder a dados RDF. Pode ser utilizada como uma "camada de dados" no RubyOnRails, facilitando a criação de aplicações Web que necessitem de lidar com RDF. Essencialmente disponibiliza uma forma de acesso programático aos modelos RDF (recursos, classes, propriedades, etc.), sem necessitar de *queries* e criando uma abstracção do repositório de informação. Esta pode ser utilizado com vários tipos de *triplestores* (Jena, Sesame, etc.) ou sobre base de dados relacionais (ex: SQLite3 [133]).

4.4 Implementação do protótipo

Tendo sido apresentada a arquitectura proposta recomendada, bem como as principais tecnologias associadas, passa-se à descrição do protótipo. Descreve-se nas várias secções a motivação e âmbito da sua construção, detalhes sobre a forma de construção do modelo e povoamento com fontes de informação conhecida, detalhes relativos à implementação da aplicação Web e sua interface, finalizando com descrição de algumas das explorações do modelo implementadas por este protótipo.

4.4.1 Âmbito e Motivação

Conforme já mencionado anteriormente, pretende-se que o foco da dissertação seja a prova de conceito da possibilidade de especificação de um modelo computável, não necessariamente a implementação de um sistema completo de suporte ao mesmo (algo que ultrapassaria o âmbito e tempo da dissertação). Com base nisto, optou-se por reduzir o âmbito do protótipo, focando-o não numa implementação total da arquitectura proposta atrás, mas nos aspectos que se consideram mais relevantes:

- A **construção do modelo proposto**, ou seja, se é possível a implementação real do mesmo na forma proposta, se é possível o povoamento e extracção de informação, e se é possível dar resposta às explorações base previstas;
- O **povoamento de informação** do modelo, com fontes já identificadas, permitindo validar a utilidade do modelo e a capacidade de produção de explorações úteis sobre esta informação. Focam-se principalmente as fontes respeitantes ao CdC (secção 3.4.1);
- A **construção de algumas interfaces** de definição de conceitos, principalmente as relativas aos CdC, UC e instância de UC. Isto permite referir alguns dos aspectos de usabilidade e interacções com o modelo, que são igualmente relevantes no apoio à definição curricular;
- A construção de algumas operações daquilo que se referiu atrás como a **componente do motor de "inferência"**. Pretende-se que estas

demonstrem a relevância de um mecanismo deste tipo no apoio à utilização do modelo, e na promoção da melhoria contínua da informação do mesmo.

- A **implementação de alguns tipos de explorações** sobre a informação existente, focada nos dados CdC e UC, demonstrando a utilidade de relatórios tabulares, gráficos, textuais, etc.

Um outro aspecto importante acerca da motivação da construção do protótipo é que não se pretendia apenas ter uma base de informação estática (ex: um EXCEL) para a produção de relatórios, mas sim testar a sua implementação em algo aproximado à arquitectura proposta.

4.4.2 Construção do modelo

Como forma de seguir a lógica base de construção do modelo, principalmente os aspectos relativos à noção de conceito, atributos, relações e extensibilidade, rapidamente se percebeu que faria sentido investigar tecnologias que utilizassem estas noções de base. A investigação dos trabalhos italianos [60] [61] e a sua modelação de conhecimento CS como ontologias, permitiu ter alguma base para o explorar de RDF, RDFS e OWL.

Conhecendo as capacidades de modelação do OWL, nomeadamente a utilização de todas as características de RDF Schema, que permite suportar a modelação de classes e hierarquias de classes, bem como a capacidade de declarar propriedades de dados ou de relacionamento (*owl:DatatypeProperty* e *owl:ObjectProperty* [66]), considerou-se que esta linguagem seria o ideal para a modelação do modelo proposto.

Iniciou-se assim a construção de uma ontologia OWL para representação do modelo proposto, seguindo os detalhes da secção 3.3 e as regras representadas na Figura 4.6, tendo especial atenção à direcção das relações e eventuais necessidades de declaração de regras de relações simétricas.

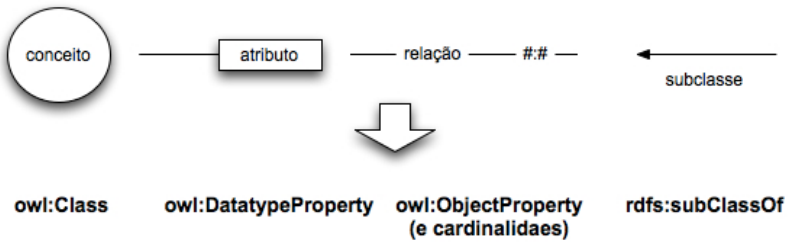


Figura 4.6 - Regras de mapeamento do modelo em OWL

Iniciou-se o trabalho de modelação, com recurso ao editor Protégé, com um foco no CdC ACM-A, definindo-se uma primeira ontologia, à qual se deu o *namespace* "*http://www.di.fct.unl.pt/madcc/acm-apx-A*" com o prefixo *acmA* e o nome *acm-apx-A*. Nesta ontologia estão modelados todos os conceitos, atributos

e relações representados na Figura 3.5, conforme se mostra parcialmente numa captura de ecrã do Protégé na Figura 4.7.

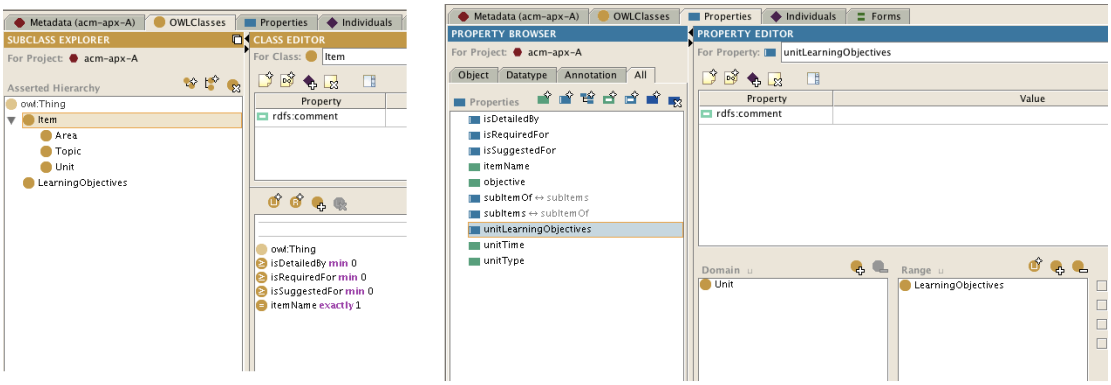


Figura 4.7 – Ontologia *acm-apx-A* no Protégé (parcial)

Refere-se para o anexo E (*acm-apx-A.owl*), onde reside o ficheiro OWL que representa esta ontologia.

Em seguida procedeu-se à modelação dos conceitos CdC ACM-B, iniciando uma nova ontologia à qual se deu o namespace "*http://www.di.fct.unl.pt/madcc/acm-apx-B*" com o prefixo *acmB* e o nome *acm-apx-B*. Naturalmente fez-se uso da propriedade OWL *owl:imports*, que permite "importar" ontologias para utilização da ontologia actualmente em modelação (entenda-se como a capacidade de referir conceitos já modelados noutra ontologia). Nesta ontologia estão modelados todos os conceitos, atributos e relações representados na Figura 3.7, conforme se mostra parcialmente, novamente numa captura de ecrã do Protégé, na Figura 4.8. Chama-se a atenção para as classes e propriedades com prefixo "acmA", que foram definidas na ontologia *acm-apx-A*.

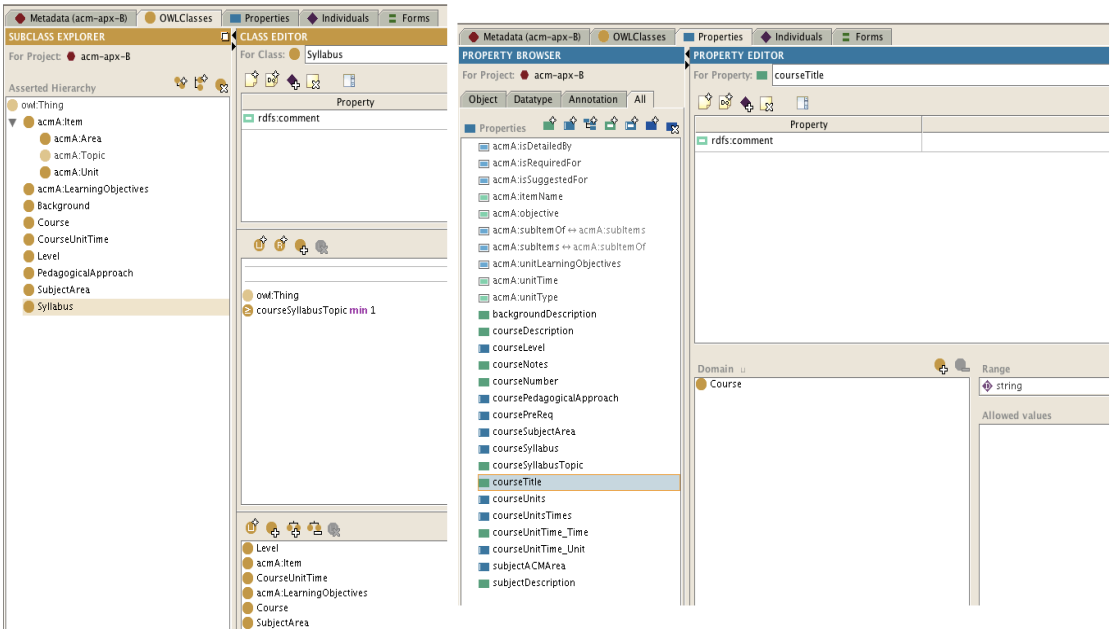


Figura 4.8 - Ontologia *acm-apx-B* no Protégé (parcial)

Refere-se para o anexo F (*acm-apx-B.owl*), que corresponde ao ficheiro OWL desta ontologia.

Passou-se então à modelação dos restantes conceitos, optando-se na altura por considerar apenas mais uma ontologia, agregadora de todo o modelo. Para esta nova ontologia, que naturalmente importa as ontologias *acm-apx-A* e *acm-apx-B,* definiu-se o namespace "*http://www.di.fct.unl.pt/madcc/di"* referindo-se apenas com o nome *DI*, por modelar conceitos implementados no âmbito do DI. Refere-se que este é um aspectos, identificado durante a escrita da dissertação, que consideramos que deveria ser revisto. Em particular a criação subdivisões de ontologias conforme o seguinte padrão: *i*) *acm-apx-A*; *ii*) *acm-apx-B*; *iii*) *skill*; *iv*) UC; *v*) Instância UC; *vi*) EC, PC e OC.

Na construção desta ontologia, é importante referir alguns aspectos aos quais se teve de prestar particular atenção, nomeadamente nas questões de hierarquias de classes e o seu impacto na modelação, bem como o reaproveitamento de relações e atributos destas. Enunciam-se os mais relevantes de seguida.

A indicação da **extensão de *di:skill* por *acmA:learningObjectives***. Essencialmente está-se a estender a definição da classe, definida atrás na ontologia acm-apx-A, indicando que a mesma é efectivamente uma subclasse da class *di:skill*. Note-se que isto está localizado na ontologia DI especificamente, não existindo qualquer alteração à acm-apx-A. Esta extensão "obrigou" igualmente a rever as propriedades OWL de acmA:learningObjectives, acrescentando novas de acordo com as relações e atributos representadas na anterior Figura 3.8. Mais simples, mas igualmente relacionado, é o facto da modelação di:curricularUnit ser directamente subclasse de acmB:course, e posteriormente a modelação de *di:curricularUnitInstance* ser subclasse de *di:curricularUnit*. Igualmente nas relações, modeladas em OWL como propriedades, e capazes de considerar extensões também devido ao mecanismo RDFS de *rdfs:subPropertyOf*, foi necessário ter atenção. Em particular refere-se a Figura 3.17 e as propriedades *di:adHocLearningObjectives* (que estende *acmA:unitLearningObjectives*) e *di:adHocSubItems* (que estende *acmA:subItems*).

Por fim, um aspecto igualmente importante é o facto de ter se tomado especial atenção em usar formas de modelações OWL apenas da linguagem OWL-Lite ou OWL-DL [134] (ex: cardinalidades "*min 0, min 1, exactly 1*"), evitando assim que a modelação fosse apenas possível em OWL-*Full*. O objectivo é deixar possível a aplicação de mecanismos de inferência sobre a ontologia.

Para ter uma noção da dimensão, veja-se a figura X novamente com uma captura de ecrã do Protégé, mostrando parcialmente as classes e propriedades (*data* e *object*) definidas. Refere-se para o anexo G (*di.owl*), onde reside o ficheiro OWL que representa esta ontologia.

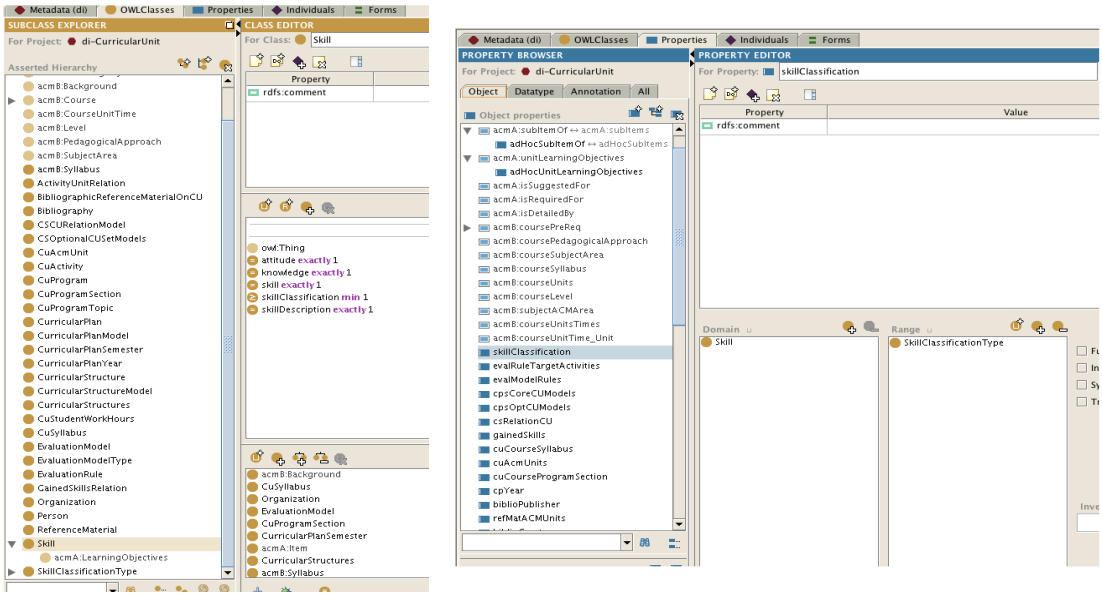


Figura 4.9 - Ontologia *DI* no Protégé (parcial)

4.4.3 Povoamento do modelo

Existindo o modelo construído como uma ontologia OWL, é importante referir que esta é ainda um documento RDF, mas que utiliza as linguagens RDFS e OWL, podendo ser representado com as várias sintaxes RDF/XML, N3, etc. Sendo a linguagem OWL uma extensão de RDFS, mais especifica para a definição de classes, propriedades, etc., esta define a **estrutura** do modelo e subsequentemente uma "linguagem" para triplos RDF. Relembrando a noção da programação orientada a objectos, e a especificação OWL [66], os **dados** existirão como instâncias das classes definidas, o que significa que são também triplos RDF. Assim, fica assente que os dados poderão também ser alimentados na ontologia como a forma de triplos RDF, com base na linguagem definida pelas ontologias OWL.

Às fontes de informação identificadas na secção 3.4.1, efectuou-se um trabalho de análise para considerar o seu povoamento automático no modelo. Dependendo dos formatos em que os dados destas existem, foi considerado o seu processamento e introdução. Percorre-se assim as várias fontes, identificando o porquê da opção do seu povoamento, bem como os vários passos efectuados na transformação e dificuldade encontradas.

4.4.3.1 Povoamento CC2001 (ACM-A e ACM-B)

De imediato se considera que o CC2001 é uma excelente base de trabalho, em particular a informação no Apx-A e Apx-B, que conterá dados para povoar *Area, Unit, Topic, Course e LearningObjectives*. Após alguma investigação, foram encontrados os apêndices A e B do CC2001 como ficheiros HTML no site do "*ACM Special Interest Group on Computer Science Education*" [28] [30] [55], algo que já não está disponível no momento de escrita da dissertação (desconhece-se o

motivo). Com base neste site, foi possível efectuar um *download* completo dos dados Apx-A e Apx-B, num formato mais manipulável, do que PDF.

Após o *download*, fizeram-se algumas operações de limpeza e transformação dos HTML para produzir ficheiros RDF com instâncias das classes declaradas anteriormente: *i*) Limpeza dos ficheiros HTML para XHTML válido e processável, com recurso à ferramenta *Tidy* [135]; *ii*) definição de um formato XML intermédio, mais simplificado para o ACM-A e ACM-B; *iii*) definição e aplicação de um XSLT [136] para transformação do XHTML no formato XML intermédio; *iv*) definição e aplicação de um XSLT para transformação do XML de formato intermédio em ficheiro com triplos RDF com a sintaxe RDF/XML. Naturalmente, e apesar do esforço, isto permitiu obter um Bok ACM CC2001 completo em forma de instancias RDF das classes da ontologia *acm-apx-A*, garantindo assim um preenchimento quase completo da mesma. O ficheiros XSLT utilizados, por se considerarem eventualmente úteis a outros, estão nos anexos H, I e J.

4.4.3.2 Povoamento Relações Extra ACM-A (*HyperKrep*)

O povoamento CC2001 não é no entanto suficiente para completar o CdC ACM-A e B, dado que se aproveitaram algumas das noções do trabalho "*Hyperkrep academic comunities*" [61] na modelação.

Sabendo porém, que este trabalho produziu estes dados, contactou-se um dos seus autores, a Dr.ª Laura Anna Ripamonti, perguntando-se qual a possibilidade de os obter. A resposta foi positiva, algo que se agradece efectivamente pela gentileza e partilha, tendo-se obtido assim um ficheiro OWL com vários dados para processar e limpar.

Dado que o ficheiro OWL vinha no formato RDF/XML, efectuou-se assim um conjunto de operações para extracção das ligações específicas (em concreto, *isSugestedFor, isRequiredFor, isDetailedBy*): criou-se um XSLT que processando o OWL da Ontologia acm-apx-A, ao mesmo tempo comparava-o com o ficheiro OWL recebido e produzida um novo ficheiro OWL acm-apx-A "aumentado" com as novas relações.

4.4.3.3 Povoamento *skills*

Considerando que o ACM-A já contém a lista de *learningObjectives*, obtêm-se já algumas *skills.* Adicionalmente, os DD e DD-e já foram anteriormente modelados como *skillsClassificationType*.

Inicialmente, dado a não existência de classificações nos learningObjectives, efectuou-se um processo automática de identificação dos seus tipos, conforme já referido anteriormente, tipos *DD_e-A* e *DI_TechSkill*. Estabeleceu-se também manualmente as relações existentes entre DD-e e DD's. Porém, conforme se pode verificar no trabalho DD-e, existem um conjunto de *skills* associadas a cada tipo de DD-e que ainda não tinha sido introduzidas. Efectuou-se portanto esse processo manualmente, atribuindo-lhe a nomenclatura "DD_e-<LetraDoTipo>-<Número>", respectivos tipos e atributos dK, dS, dA.

Finalmente, e de acordo com a informação do trabalho do DI "Proposta de curso de 1º ciclo de Licenciatura em Engenharia Informática" [71], introduziram-se manualmente um conjunto de *skills* técnicas e *soft-skills* já identificadas no documento, igualmente fazendo a sua classificação manual com os tipos respectivos e atributos dK, dS e dA.

Adicionalmente, os aspectos de alimentação da informação de *learningObjectives* com as suas relações para Topics e verbos Bloom, não foram efectuados aqui neste povoamento inicial. Pensou-se que utilização da componente de descoberta de relações (ou motor de "inferência") poderia ser útil na proposta de informação, pelo que se abordará este aspecto mais à frente.

4.4.4 Implementação da aplicação *Web*

Tendo então uma ontologia OWL definida, bem como um povoamento aceitável para o CdC, igualmente como instâncias RDF. Foi possível iniciar-se a construção do protótipo como uma aplicação *Web* exemplo. O objectivo desta secção é descrever os aspectos de construção desta aplicação, focando-se apenas naquilo que se considerou mais relevantes (refere-se o descrito no início deste capítulo). Considera-se importante iniciar esta descrição dizendo que se optou por não implementar questões de multi-utilizador, perfis, versionamento e histórico, dado ser algo que está fora do âmbito definido para o protótipo.

Utilizando-se as noções do *RubyOnRails* e *ActiveRDF*, referidas atrás, pode-se resumir a sua arquitectura aos seguintes componentes:

- **Camada de Dados** – Uma base de dados Saline, existente portanto como um ficheiro ".sqlite" e utilizado pelo ActiveRDF;
- **Camada de acesso aos dados** – Utiliza-se o *ActiveRDF*. Este utiliza o ficheiro SQLite acima referido, ou caso não exista, efectua o carregamento inicial do ficheiro SQLite com base num documento de triplos RDF em formato N3. Naturalmente, para *setup* inicial converteram-se os ficheiros OWL das ontologias e instâncias para um único ficheiro NT que foi então gerado num SQLite pelo ActiveRDF. São aqui disponibilizados objectos base para aceder aos dados RDF;
- **Camada de objectos "lógicos"** – utilizam-se *Models* que aproveitam o ActiveRDF e estendem-no com mais operações se necessário;
- **Camada de controlo** – utilizam-se *Controllers* que definem o conjunto de operações base para os vários elementos da aplicação, resultando tipicamente na visualização de uma interface;
- **Camada interface** – utilizam-se *Views*, bem como HTML e Javascript para produzir uma interface *Web;*
- **Camada de relatórios** – Utilizam *Controllers, Views,* para gerar Tabelas e HTML, e outras bibliotecas específicas para gerar Gráficos (Barras, etc.) e Grafos, respectivamente *Grude* e uma combinação RGL + graphViz;

- **Camada de "inferência"** – não sendo propriamente uma "zona" à parte, utiliza *Controllers* para definir algumas operações sobre o modelo (descritas à frente) que utilizam as bibliotecas *Classifier* (classificador Bayesiano genérico e *latent sematic indexing*), *Ferret* e *Stemmer*.

Tendo este *overview*, detalham-se alguns aspectos que se consideram mais interessantes na interface desenvolvida. Focou-se este protótipo em dois aspectos principais, a possibilidade de consulta do CdC e a edição / consulta de UC (apesar de se terem implementado algumas interfaces de edição de EC, PC e OC). Mostram-se então algumas capturas de ecrãs focadas nestes aspectos.

Relativamente à consulta do CdC, em particular do ACM-A, dá-se alguns exemplos: *i*) Na Figura 4.10, uma interface de consulta assistida à navegação do CdC, onde é possível percorrer todo o ACM-A, navegando nas áreas e seus filhos (Units, Topics, etc.), disponibilizando inclusive uma pesquisa de texto livre*; ii*) Na Figura 4.11, uma apresentação da AREA "*Information Management (IM)"* e suas *Units*, com a visualização parcial do seu grafo de relações, gerado automaticamente pelo sistema; *iii*) Na Figura 4.12, uma visão da acmA:Unit "*SE2. Using APIs*", onde é possível verificar os seus *Topics*, *LearningObjectives* e relações adicionais de *Item.* Relativamente à componente ACM-B, mostra-se simplesmente na Figura 4.13 a visualização parcial de um *acmB:course "CS112F. Objecte and Algorithms".*

Repare-se na presença de *links* para permitir a navegação em praticamente todos os conceitos e subconceitos da interface.

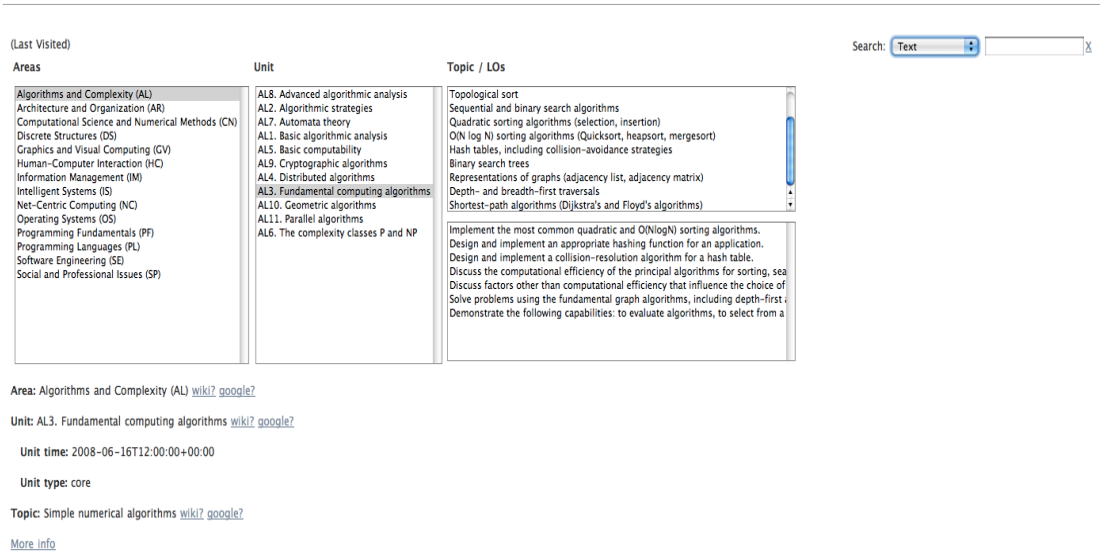


Figura 4.10 - Navegação e consulta do CdC - ACM-A

Area: http://www.di.fct.unl.pt/scqce/acm-apx-A#Area_IM

Name: Information Management (IM)

SubItems (Units):

- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-DataMining
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-DataModeling
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-DatabaseQueryLanguages
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-DatabaseSystems
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-DigitalLibraries
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-DistributedDatabases
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-Hypermedia
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-InformationModels
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-Multimedia
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-Physical
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-RelationalDatabaseDesign
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-RelationalDatabases
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-StorageRetrieval
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IM-TransactionProcessing

isSuggestedFor:

- http://www.di.fct.unl.pt/scqce/acm-apx-A#Topic_SP-PrivacyCivilLiberties_2 – Privacy implications of massive database systems
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_NC-WebApplications – NC5. Building web applications

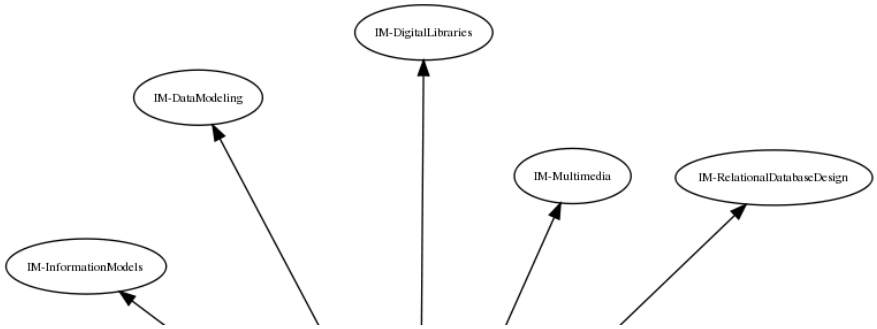


Figura 4.11 – Visualização de uma acmA:AREA

Unit: http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_SE-UsingAPIs

Name: SE2. Using APIs

SubItemOf (Area): http://www.di.fct.unl.pt/scqce/acm-apx-A#Area_SE

Unit time: 2008-06-16T05:00:00+00:00

Unit type: core

SubItems (Topics):

- http://www.di.fct.unl.pt/scqce/acm-apx-A#Topic_SE-UsingAPIs_1 – API programming
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Topic_SE-UsingAPIs_2 – Class browsers and related tools
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Topic_SE-UsingAPIs_3 – Programming by example
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Topic_SE-UsingAPIs_4 – Debugging in the API environment
- http://www.di.fct.unl.pt/scqce/acm-apx-A#Topic_SE-UsingAPIs_5 – Introduction to component-based computing

SubItems (LearningObjectives):

- http://www.di.fct.unl.pt/scqce/acm-apx-A#LO_SE-UsingAPIs_1 – Explain the value of application programming interfaces (APIs) in software development.

  Verb: Explain

  Topic: http://www.di.fct.unl.pt/scqce/acm-apx-A#Topic_SE-UsingAPIs_1 – API programming

- http://www.di.fct.unl.pt/scqce/acm-apx-A#LO_SE-UsingAPIs_2 – Use class browsers and related tools during the development of applications using APIs.

  Verb: Use

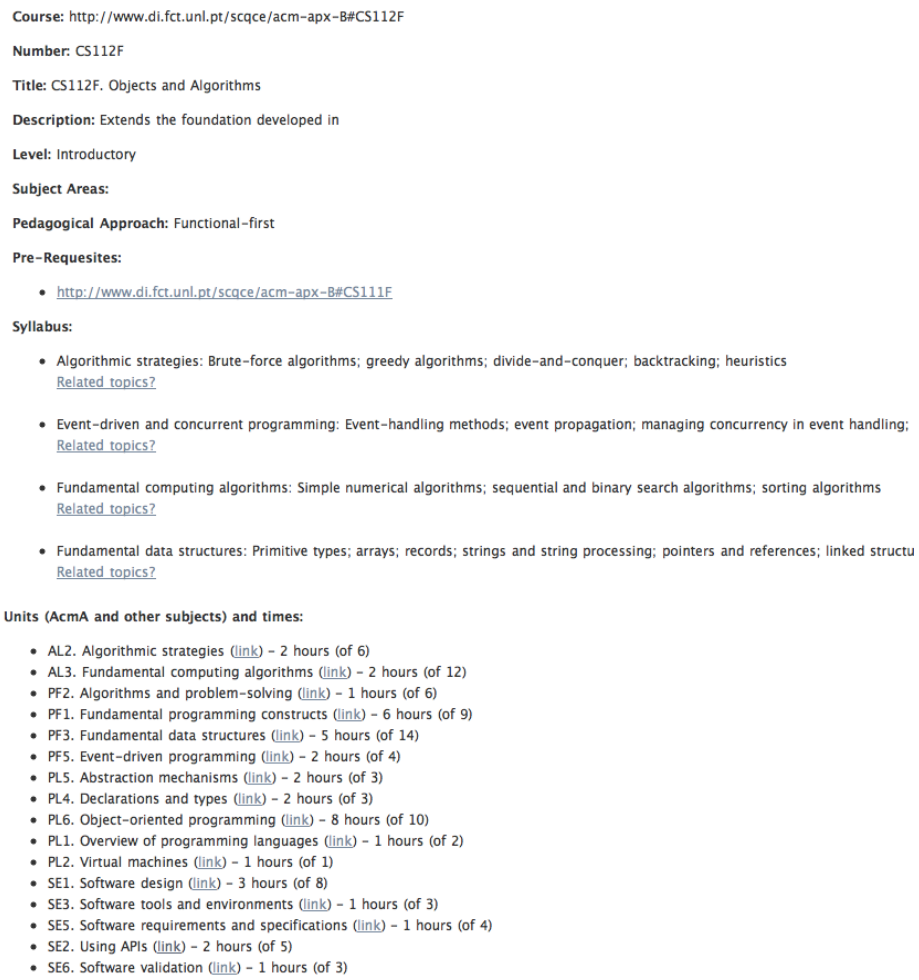
  Topic: http://www.di.fct.unl.pt/scqce/acm-apx-A#Topic_SE-UsingAPIs_2 – Class browsers and related tools

- http://www.di.fct.unl.pt/scqce/acm-apx-A#LO_SE-UsingAPIs_3 – Design, implement, test, and debug programs that use large-scale API packages.

  Verb: Design,

  Topic: http://www.di.fct.unl.pt/scqce/acm-apx-A#Topic_SE-UsingAPIs_1 – API programming

Figura 4.12 – Visualização de uma acmA:Unit



Course: http://www.di.fct.unl.pt/scqce/acm-apx-B#CS112F

Number: CS112F

Title: CS112F. Objects and Algorithms

Description: Extends the foundation developed in

Level: Introductory

Subject Areas:

Pedagogical Approach: Functional-first

Pre-Requesites:

- http://www.di.fct.unl.pt/scqce/acm-apx-B#CS111F

Syllabus:

- Algorithmic strategies: Brute-force algorithms; greedy algorithms; divide-and-conquer; backtracking; heuristics
  Related topics?
- Event-driven and concurrent programming: Event-handling methods; event propagation; managing concurrency in event handling;
  Related topics?
- Fundamental computing algorithms: Simple numerical algorithms; sequential and binary search algorithms; sorting algorithms
  Related topics?
- Fundamental data structures: Primitive types; arrays; records; strings and string processing; pointers and references; linked structu
  Related topics?

Units (AcmA and other subjects) and times:

- AL2. Algorithmic strategies (link) - 2 hours (of 6)
- AL3. Fundamental computing algorithms (link) - 2 hours (of 12)
- PF2. Algorithms and problem-solving (link) - 1 hours (of 6)
- PF1. Fundamental programming constructs (link) - 6 hours (of 9)
- PF3. Fundamental data structures (link) - 5 hours (of 14)
- PF5. Event-driven programming (link) - 2 hours (of 4)
- PL5. Abstraction mechanisms (link) - 2 hours (of 3)
- PL4. Declarations and types (link) - 2 hours (of 3)
- PL6. Object-oriented programming (link) - 8 hours (of 10)
- PL1. Overview of programming languages (link) - 1 hours (of 2)
- PL2. Virtual machines (link) - 1 hours (of 1)
- SE1. Software design (link) - 3 hours (of 8)
- SE3. Software tools and environments (link) - 1 hours (of 3)
- SE5. Software requirements and specifications (link) - 1 hours (of 4)
- SE2. Using APIs (link) - 2 hours (of 5)
- SE6. Software validation (link) - 1 hours (of 3)

Figura 4.13 – Visualização (parcial) de um acmB:course

Por fim, mostra-se a interface de consulta/edição de uma UC, onde é possível por exemplo, definir as *UNITs* a abordar e detalhar os seus ECTS (Figura 4.14), definir tópicos não cobertos (Figura 4.15) ou detalhar uma *adHocUnit* (Figura 4.16), definir *skills* (Figura 4.17), entre outros. Refere-se em particular os aspectos de apoio à definição na *interface* de edição, por exemplo a Figura 4.18, onde se permite na edição de *Units* ter acesso a uma zona de consulta assistida ao CdC. (conforme visto atrás).

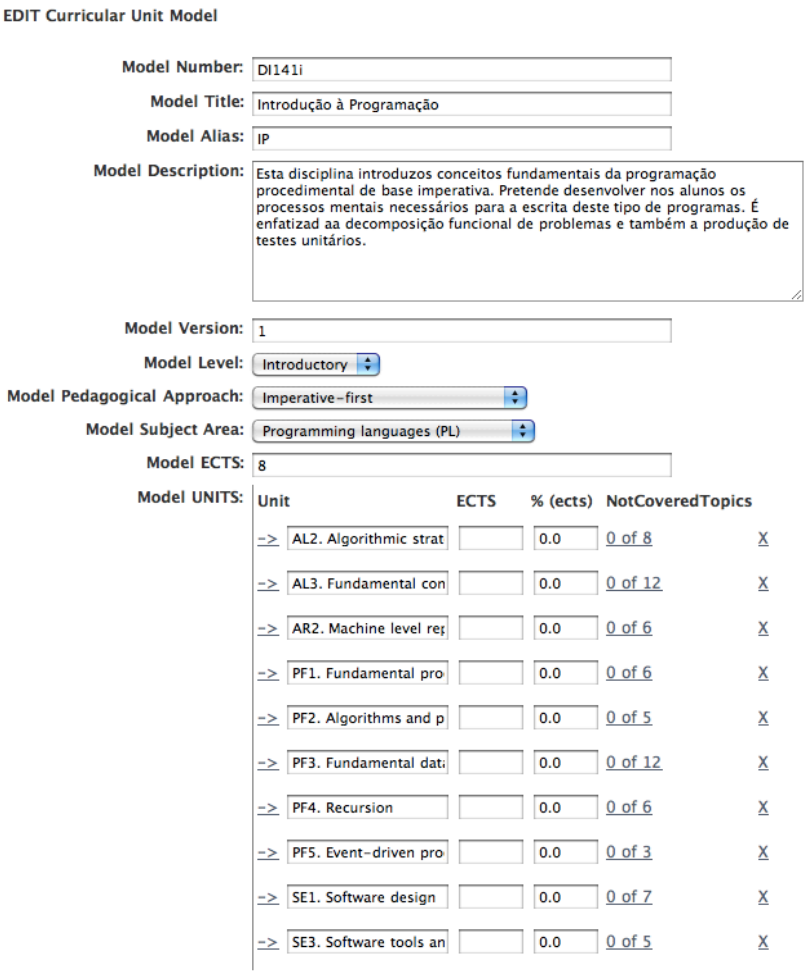


Figura 4.14 – Edição de uma UC (parcial com foco em Units)

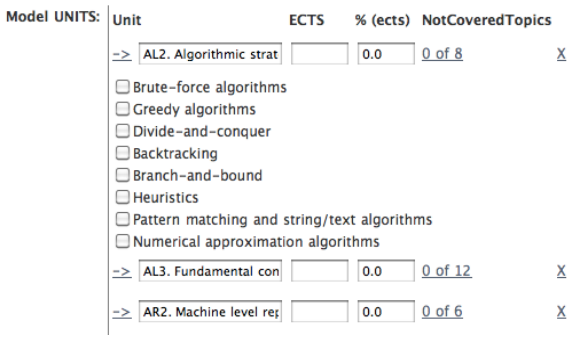


Figura 4.15 - Edição de UC - definição de UNITS e tópicos não cobertos

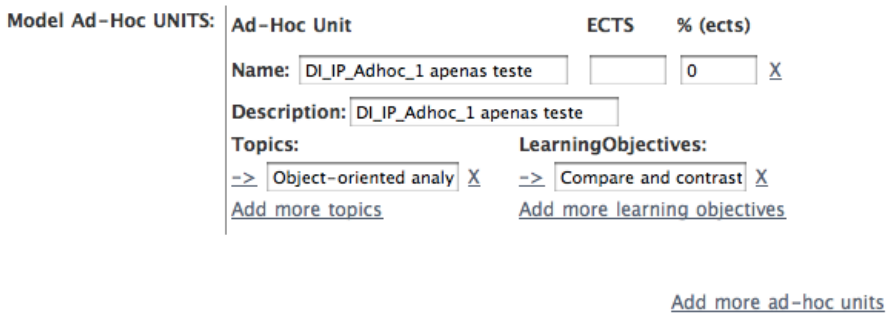


Figura 4.16 - Edição de UC - definição de AdHocUNITS

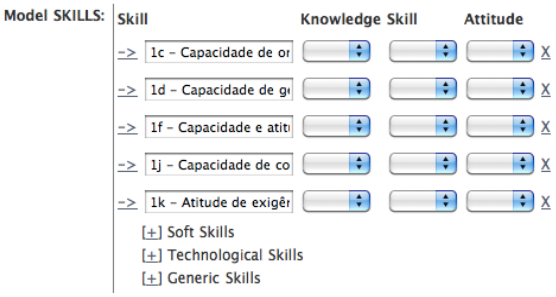


Figura 4.17 – Edição de UC – definição de *Skills*

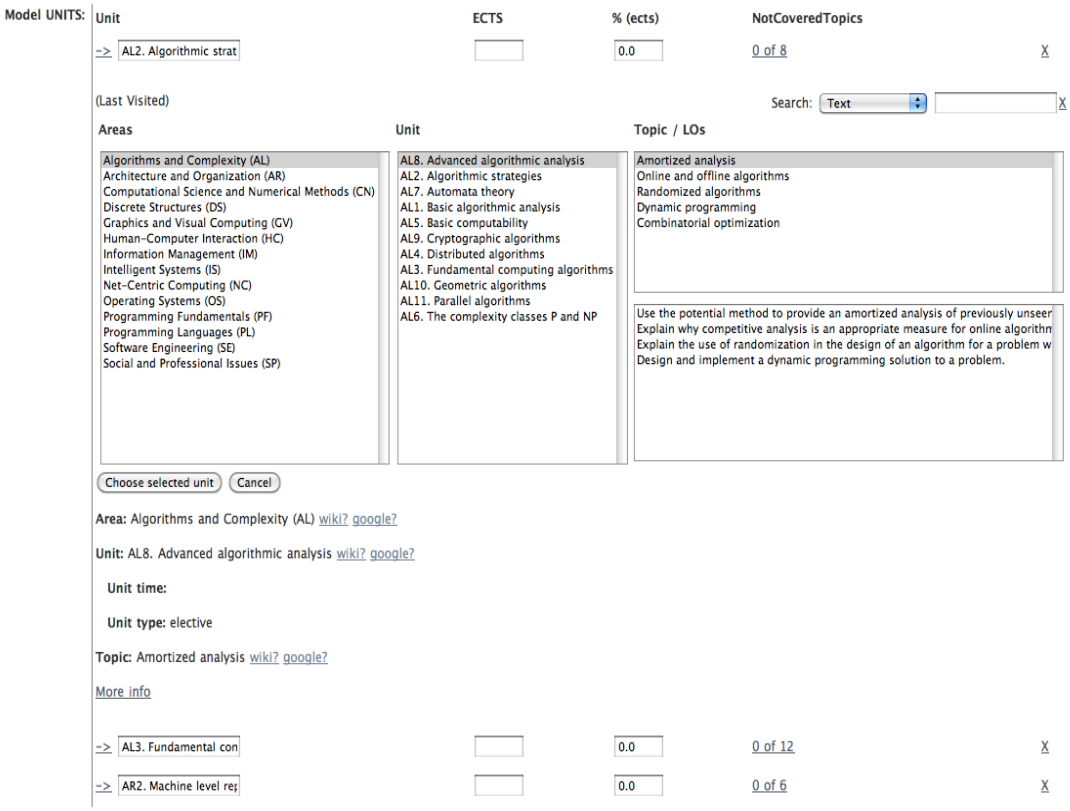


Figura 4.18 – Edição de UC – apoio à definição de UNITS – navegação no CdC

Alguns aspectos relativos aos relatórios e aos mecanismos de descoberta de novas relações, optou-se por descrever na secção seguinte.

4.4.5 Exploração do protótipo

Dado a implementação do modelo, principalmente do CdC, da existência desta arquitectura e suas componentes de apoio, foi possível extrair algumas explorações que se julgam interessantes e consistentes com o pretendido atrás para o modelo.

Primeiramente descrevem-se os aspectos relativos a mecanismos de descoberta de novas relações (aquilo que perfaz a implementação mínima da componente motor de inferência do protótipo), seguindo-se para alguns relatórios exemplo desenvolvidos no sentido de mostrar o que é possível com o modelo.

4.4.5.1 Descoberta de novas relações

Já foi referido anteriormente, na secção 4.4.3.3 relativa ao povoamento de *Skills*, que não se estabeleceram algumas das novas relações propostas para os *learningObjectives*, em concreto as relacionadas com a identificação dos *Topics* que lhe deram origem e o respectivo Verbo da Taxonomia de Bloom. Isto foi propositado, pois pela análise feita aos textos dos LO's durante a construção do modelo, e a sua aproximação com os textos dos *Topics*, suspeitou-se que era possível estabelecer este relacionamentos de forma automática ou automaticamente assistida.

Isto relevou-se interessante para o desenvolvimento dentro da componente de "inferência" e descoberta de relações, comprovando assim a sua utilidade na arquitectura proposta. Sabendo que se estava a lidar principalmente com textos, optou-se por investigar alguns algoritmos de classificação automática de texto, tendo descoberto uma biblioteca *Ruby* (*Classifier*) que implementava um *Naive Bayes classifier* [103] e *Latent semantic indexing* [104]. Com base nisto, e numa análise transversal aos vários textos dos LO's, definiu-se a seguinte estratégia, para todas *Units*:

- Percorrer todos os seus *Topics* e:
  - Aplicar ao algoritmo *Naive Bayes,* um conjunto de treino que associa como categoria o nome da *unit* ao texto do *topic*;
  - No algoritmo *Latent semantic indexing*, adicionar o texto do topic, construindo um índice no final (de todas as units);
- Percorrer novamente todos os *topics* de todas as *units e:*
  - Utilizando o algoritmo LSI, perguntar se existem tópicos relacionados com o texto deste tópico, guardando a sua lista;
- Percorrer todos os seus LO's e:
  - Assumir que o Verbo da Taxonomia de Bloom é tipicamente a primeira palavra do texto dos LO's, definindo-a como tal;

Tendo isto, é possível então elaborar diferentes operações para dinamicamente na aplicação se:

- Encontrar os Topics relacionados com determinado Topic (via texto) – Ver Figura 4.19;

- Encontrar o primeiro *Topic* mais relacionado com o texto do LO – Ver Figura 4.20;
- Encontrar os *Topic* relacionados com os textos de cada ponto do Syllabus de um *course* – Ver Figura 4.21*;*
- Encontrar *Units* para um *course* apenas com base no syllabus – Ver Figura 4.22;
- Estipular automaticamente assistida a atribuição dos verbos de Bloom – Ver Figura 4.20;

Topic: http://www.di.fct.unl.pt/scqce/acm-apx-A#Topic_NC-NetworkSecurity_1

Name: Fundamentals of cryptography

SubItemOf (Unit): http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_NC-NetworkSecurity

The text "Fundamentals of cryptography" may relate to the following topics:

- Fundamental definitions – http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_IS-FundamentalIssues
- Fundamentals – http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_SE-ComponentBased
- Public-key cryptography – http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_AL-Cryptography

Figura 4.19 – *Topics* relacionados com o *Topic*

SubItems (LearningObjectives):

- http://www.di.fct.unl.pt/scqce/acm-apx-A#LO_AL-BasicComputability_1 – Discuss the concept of finite state machines.

  Verb: Discuss

  Topic: http://www.di.fct.unl.pt/scqce/acm-apx-A#Topic_OS-Concurrency_1 – States and state diagrams

- http://www.di.fct.unl.pt/scqce/acm-apx-A#LO_AL-BasicComputability_2 – Explain context-free grammars.

  Verb: Explain

  Topic: http://www.di.fct.unl.pt/scqce/acm-apx-A#Topic_AL-BasicComputability_2 – Context-free grammars

Figura 4.20 - *Topic* mais provável relacionado com o texto do LO

Course: http://www.di.fct.unl.pt/scqce/acm-apx-B#CS102O

Number: CS102O

Title: CS102O. Objects and Data Abstraction

Description: Continues the introduction from CS101O to the methodology of programming from an object-oriented perspective. Through the study interfaces, graphics, and the social implications of computing, with an emphasis on software engineering.

Level: Introductory

Subject Areas:

Pedagogical Approach: Objects-first

Pre-Requesites:

- http://www.di.fct.unl.pt/scqce/acm-apx-B#CS101O

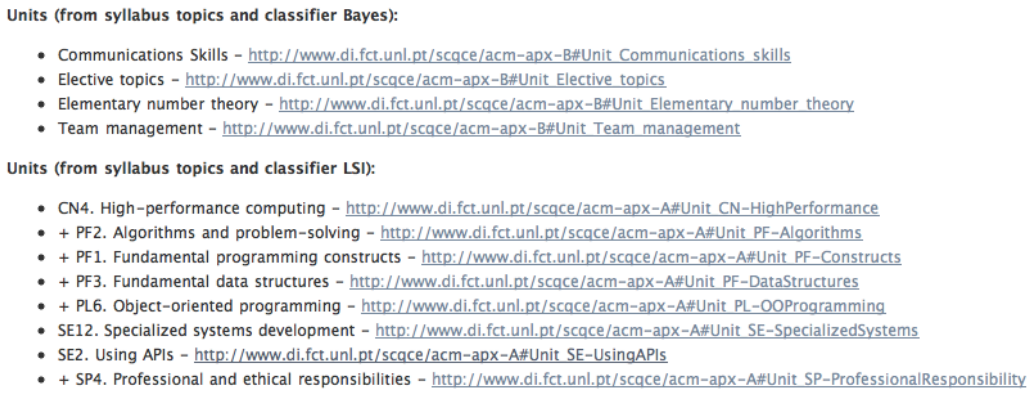
Syllabus:

- Event-driven programming: Event-handling methods; event propagation; exception handling

  The text "Event-driven programming: Event-handling methods; event propagation; exception handling" may relate to the following topics:

  - Event-handling methods – http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_PF-EventDriven
  - Event propagation – http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_PF-EventDriven
  - Exception handling – http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_PF-EventDriven

Figura 4.21 - *Topics* relacionados com os pontos do *Syllabus* de *Course*



Units (from syllabus topics and classifier Bayes):

- Communications Skills - http://www.di.fct.unl.pt/scqce/acm-apx-B#Unit_Communications_skills
- Elective topics - http://www.di.fct.unl.pt/scqce/acm-apx-B#Unit_Elective_topics
- Elementary number theory - http://www.di.fct.unl.pt/scqce/acm-apx-B#Unit_Elementary_number_theory
- Team management - http://www.di.fct.unl.pt/scqce/acm-apx-B#Unit_Team_management

Units (from syllabus topics and classifier LSI):

- CN4. High-performance computing - http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_CN-HighPerformance
- + PF2. Algorithms and problem-solving - http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_PF-Algorithms
- + PF1. Fundamental programming constructs - http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_PF-Constructs
- + PF3. Fundamental data structures - http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_PF-DataStructures
- + PL6. Object-oriented programming - http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_PL-OOProgramming
- SE12. Specialized systems development - http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_SE-SpecializedSystems
- SE2. Using APIs - http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_SE-UsingAPIs
- + SP4. Professional and ethical responsibilities - http://www.di.fct.unl.pt/scqce/acm-apx-A#Unit_SP-ProfessionalResponsibility

Figura 4.22 – Units descobertas com base nos textos do *Syllabus* de *Course*

Estas explorações consideram-se apenas alguns exemplos daquilo que poderia ser possível desenvolver na componente do motor de "inferência", demonstrando que tal poder ser útil à consulta genérica do corpo de conhecimento, bem como ao apoio à definição curricular.

4.4.5.2 Relatórios desenvolvidos

Outro tipo de exploração, mais relacionado com as referidas directamente no modelo, são os relatórios e gráficos que se pode produzir com base na informação que este contém. Construiu-se na aplicação a capacidade de gerar diferentes tipos de resultados nos relatórios, sendo no entanto necessário construir o relatório no sistema, algo que apenas requer a definição de um novo *Controles (*e dependendo do tipo, mais uma *View),* responsável pela execução das operações e outputs.

Definiram-se então, a título exemplificativo os seguintes relatórios, gerados dinamicamente pela aplicação:

- Distribuição do número de Units por Area (percentagem);
- Distribuição do número de Topics por Area (percentagem);
- Distribuição do número de LOs por Area (percentagem) – Ver Figura 4.23 ;
- Cobertura das *Units* em determinado *Course* (Barras – Figura 4.24, ou Radial – Figura 4.25);
- Matriz de distribuição do número de *Units* de cada *Area*, para cada *Course* - Figura 4.26*;*
- Um grafo com as relações AREA-UNIT para todo o CdC ACM-A (uma figura com uma dimensão não possível visualizar, mas incluído no anexo K em formato digital);

Considera-se que este tipo de componente é realmente interessante, e num sistema mais completo e com informação mais detalha relativa a UCs e ECs, poderá produzir resultados muito úteis ao apoio à definição curricular.

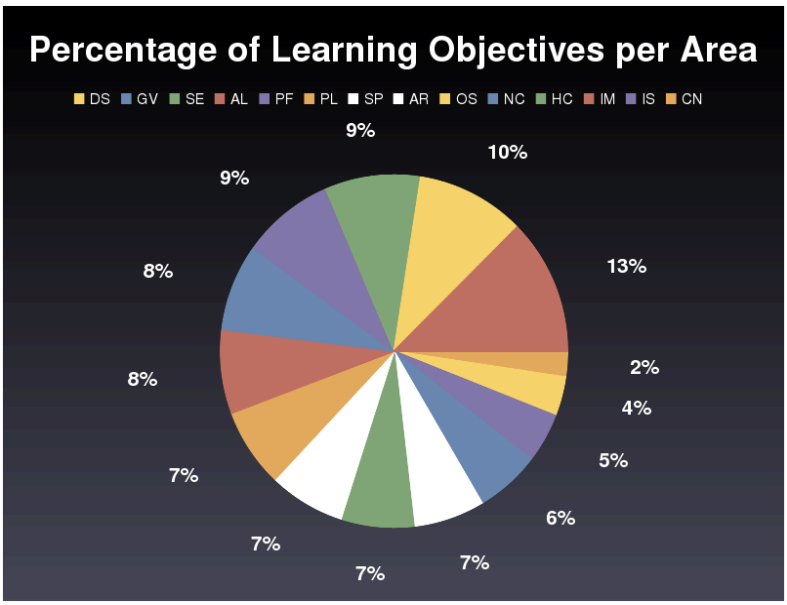


Figura 4.23 – Distribuição do número de LOs por Area (%)

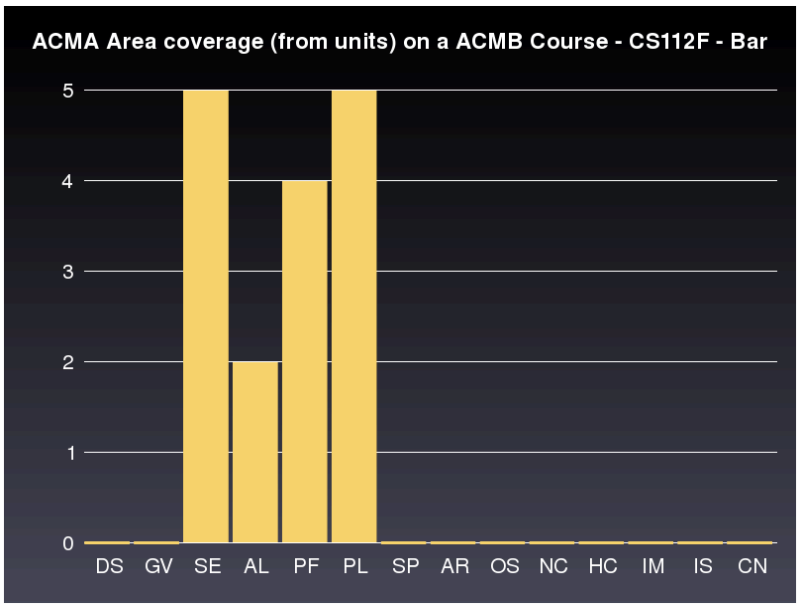


Figura 4.24 – Cobertura das Units em determinado Course (Barras)

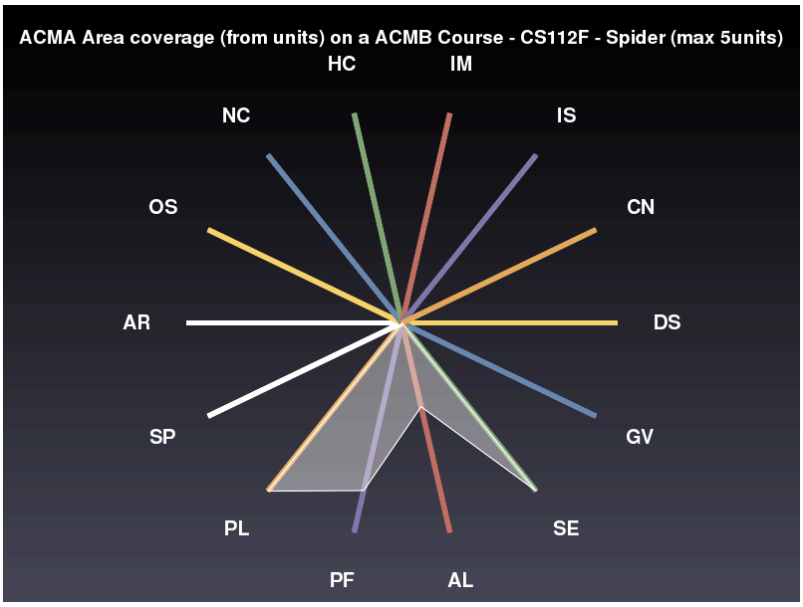


Figura 4.25 – Cobertura das Units em determinado Course (Radial)

Course	DS	GV	SE	AL	PF	PL	SP	AR	OS	NC	HC	IM	IS	CN	ZeroSeq.
CS100B	2	2	0	2	2	1	1	1	2	2	0	0	0	0	56
CS101B	5	0	0	3	4	2	2	1	0	0	0	0	0	0	45
CS101I	0	1	3	0	3	3	4	2	1	1	1	0	0	0	49
CS101O	0	0	2	2	3	4	3	1	0	0	0	0	0	0	57
CS102B	2	0	4	1	4	5	0	3	0	1	2	0	0	0	9
CS102I	0	0	5	1	4	4	0	1	0	0	2	1	0	0	7
CS102O	0	1	6	2	3	4	0	1	0	0	1	0	0	0	4
CS103B	5	0	3	2	3	3	0	0	1	0	0	0	1	0	2
CS103I	1	0	1	4	2	2	0	0	0	0	0	0	0	0	42
CS103O	0	0	2	3	3	0	0	0	0	0	0	0	0	0	33
CS105	4	0	0	0	0	0	0	1	0	0	0	0	0	0	7
CS106	4	0	0	3	0	0	0	0	0	0	0	0	0	0	11
CS111A	0	0	2	5	4	3	1	0	0	0	0	0	0	0	50
CS111F	1	0	2	4	4	4	1	0	1	0	0	0	0	0	6
CS111H	0	0	3	3	4	3	1	4	0	0	0	0	0	0	57
CS111I	0	1	4	2	3	4	1	2	0	0	0	0	0	0	70
CS111O	0	1	3	2	4	3	2	0	0	0	0	0	0	0	63
CS112A	0	1	5	3	3	6	0	0	0	0	0	0	0	0	55
CS112F	0	0	5	2	4	5	0	0	0	0	0	0	0	0	42
CS112H	0	0	4	3	4	5	0	0	0	0	0	0	0	0	42
CS112I	1	0	3	2	2	6	0	0	0	0	0	0	0	0	42
CS112O	0	0	4	3	5	5	0	0	0	0	1	0	0	0	4

Figura 4.26 – Matriz de distribuição do número de *Units* de cada *Area*, para cada *Course*

Capítulo 5
Avaliação do trabalho e Conclusões

Neste capítulo finalizam-se com 2 aspectos, uma breve avaliação do trabalho efectuado nesta dissertação, analisando-se a proposta do modelo e construção do protótipo, e elaboram-se algumas conclusões finais, introduzindo-se também a discussão de trabalho futuro.

Neste último capítulo, apresentam-se algumas considerações acerca da avaliação do trabalho, bem como aspectos relativos a trabalho futuro e conclusões finais, optando-se por o dividir em 3 secções.

5.1 Avaliação do trabalho

O foco deste trabalho é efectivamente o modelo e a demonstração que é possível a sua criação num sistema computável, trazendo isso resultados úteis para apoio à definição curricular e consultas sobre a mesma. Os aspectos mais imediatos que se podem considerar para avaliação do modelo são: *i*) Capacidade de modelar a informação pretendida; *ii*) Capacidade de extensibilidade e lidar com a evolução natural necessária; *iii*) Capacidade de dar reposta às explorações propostas.

Uma outra componente deste trabalho reside na proposta de uma arquitectura recomendada para uma implementação futura deste modelo e um protótipo com algumas das componentes já implementadas. Este protótipo considera-se que está directamente relacionado com os pontos *i* e *iii*, pelo que poderá apoiar a sua validação. Assim, definem-se aqui 2 secções, uma relativa à avaliação da extensibilidade do modelo, outra relativa à implementação do modelo no protótipo.

5.1.1 Extensibilidade do Modelo

Descreve-se algumas considerações relativas aos aspectos de extensibilidade do modelo, ou seja, a capacidade de lidar com as naturais necessidades de adição de novas "peças", ou eventuais revisões da base definida.

Apesar de, à partida, alguns destes aspectos estarem também relacionados com as diversas opções tecnológicas que poderiam ser tomadas numa eventual implementação do mesmo, o objectivo nesta secção foca-se apenas no descrever como a base conceptual do modelo lidaria com remodelações e extensões. Note-se que não se está a descrever alimentação de "novos" dados (algo tácito no modelo), mas sim introdução/revisão de conceitos.

Pode-se utilizar como base a Figura 3.2 e, alterando-a, identificar algumas das componentes base do modelo, que pelo seu nível de inter-relacionamento se consideram como "pedras base" e deverão por isso existir sempre nesta forma ou noutra similar. Tal está representado na Figura 5.1, onde estes conceitos foram agora "pintados" em rectângulos com o fundo totalmente preto.

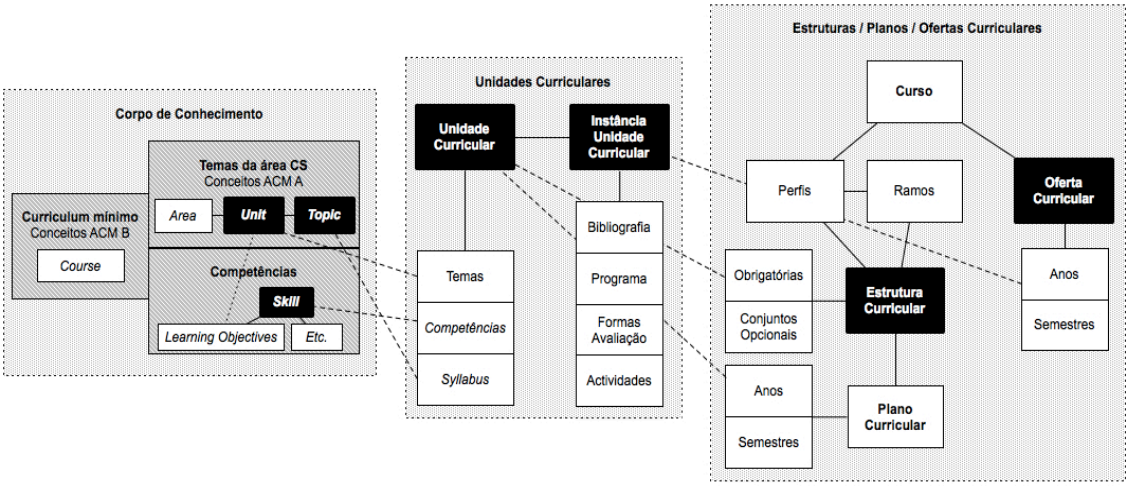


Figura 5.1 – Visão geral do modelo, com os principais conceitos que pela sua inter-dependência são "pedras base" do modelo.

É importante explicar, ou "defender", o porquê de se considerarem cada um destes conceitos como "pedras base", passíveis de existir sempre no modelo:

- *Unit* e *Topic* (vindo de ACM-A) – Este tipo de estruturas permite definir e detalhar os principais temas que os cursos abordam. São completamente relevantes para a definição da estrutura curricular das disciplinas e dos cursos, pelo que deverão existir sempre (independente dos nomes).
- *Skills* (vindo de Bolonha, ACM-A, etc.) – A identificai de competências que os cursos e disciplinas pretendem transmitir aos alunos, é algo estruturalmente importante na remodelação de Bolonha. As competências são efectivamente a base comparativa dos níveis de performance, ou sucesso dos cursos / disciplinas. Considera-se portanto que o conceito deverá existir sempre, podendo naturalmente ter diversas subformas.
- Estruturas Curriculares (EC's) e Unidades Curriculares (UC's) – são inequivocamente peças fundamentais para a estruturação curricular, que podendo eventualmente existir sobre outros nomes e com diferenças ao nível dos sub-conceitos, são algo que deverá existir sempre neste modelo. São os agrupadores dos conceitos base dos cursos.
- Instâncias de UC e Oferta Curricular (OC's) – são as operacionalizações das estruturações acima, pelo que se pretende que o modelo considere os aspectos operacionais, deverão sempre existir com formas aproximadas às actuais.

Naturalmente, dentro destes conceitos, os subconceitos que estabelecem relações entre as "pedras base" são igualmente relevantes, pelo que se considera este tipo de relacionamento (exemplos: *Units* que determinada UC deverá

abordar, *Topics* de determinada *Unit*, etc.), como a "cola" relevante do modelo, devendo sempre existir.

Tendo esta base, pode-se então descrever algumas considerações sobre as eventuais formas de extensibilidade das diversas componentes do modelo, algo que se optou por dividir em duas subsecções: *i*) introdução de novos conceitos base, quer para associação aos já existentes, quer para criação apenas de novos componentes no modelo; *ii*) extensões nos actuais conceitos, ou seja, quer também para a introdução de novos aspectos (sub-conceitos) em conceitos já existentes, ou sub-tipagem desses conceitos para marcar diferenciação.

Existe ainda uma última secção onde se incluem alguns comentários sobre as secções do modelo onde se consegue antever a necessidade de revisão ou extensão.

5.1.1.1 Introdução de novos conceitos

A introdução de novos conceitos no modelo considera-se algo possível, e perfeitamente aceitável. Naturalmente, considera-se que uma eventual implementação do modelo terá de ter este aspecto presente e lidar com a possibilidade de introdução de novos conceitos relativamente aos aspectos técnicos (revisão de interfaces, da camada de dados, etc.).

Antevê-se 2 cenários na introdução de novos conceitos:

- **Novo conceito** de "1º nível", independentemente da área conceptual, com ligações (num sentido) a conceitos já existentes. Portanto sem alteração de nenhum conceito anteriormente existentes.
  Imagine-se um exemplo fictício, a definição do novo conceito de Regente ou Professor, onde é possível serem especificados os seus temas de principal investigação ou ensino (via ligação a Topics ou Units), anteriores disciplinas leccionadas (via ligação a UC), etc. Este poderia ser utilizado para, durante a definição de uma UC (ou instância), e com recurso a um algoritmo adequado, sugerir qual o regente mais "compatível" com a estrutura conceptual da UC.

- **Novo subconcessão**, de um actual conceito existente, que justifique uma remodelação do mesmo. Portanto, considerou-se suficientemente relevante a alteração para modificar um actual conceito existente, desde que seja garantido que isto não afecta as restantes ligações do modelo. Isto consiste em introduzir um novo subconceito, que esteja directamente relacionado com determinado conceito existente, o que obriga a introduzir uma nova relação (ou relações) neste.
  Um exemplo fictício, pode ser a introdução nas *Unit* de indicação de referências bibliográficas de suporte às mesmas. É algo que pode justificar fazer a alteração ao conceito *Unit*, desde que se garanta que nenhuma das outras ligações são afectadas.

5.1.1.2 Extensão nos actuais conceitos

Tem até aqui sido assumido como base a característica ou noção de extensão / herança nos vários conceitos do modelo, algo que uma eventual abordagem à implementação deste modelo deverá necessariamente suportar (seja ela, de que tipo for). Esta capacidade de extensão dos conceitos, que foi usada na modelação de algumas situações (ex: *Skills* e definição dos seus tipos e subtipos, *acmA:LearningObjectives* como extensão de *Skills*, etc.), é algo que se pensa permitir lidar com as evoluções mais complexas dos actuais conceitos.

Considera-se que, para efeitos deste modelo, esta noção de extensão permite, além da herança de relações e atributos, redefinir o comportamento dos mesmos. Naturalmente, é necessário garantir ao nível da lógica do modelo, a coerência nas heranças e respectivos conceitos sub-tipados, em particular nas relações obrigatórias de ligação entre os conceitos "base". Isto é da responsabilidade de quem adiciona novos conceitos ao modelo (e naturalmente, de quem for responsável por uma eventual implementação).

Antevê-se 3 cenários para a utilização da extensão:

- **Novo subconceito**, de um actual conceito existente, em situações onde não é desejável alterar o mesmo. Uma situação relacionada com a descrita acima, onde surgindo a necessidade de "rever" um conceito e acrescentar relações para novos subconceitos, verificou-se que tal não é desejável (por exemplo, por retro compatibilidade com outra "zonas" do modelo).

  Nesta situação, considera-se efectuar um novo conceito, subtipo do anterior, onde se poderá acrescentar novas relações para o novo subconcessão criado e eventualmente (caso o modelo o permita), redefinir relações herdadas.

- **Hierarquia de conceitos ou introdução de "tipos de conceito"**, são situações onde é necessário "partir" efectivamente o conceito em diferentes subtipos, sendo desejável estipular directamente no modelo essas distinções.

  Refere-se a situação da modelação do tipo de *skills*, ou por exemplo, uma situação fictícia onde poderá ser necessária efectuar uma distinção entre diferentes tipos de materiais de referência de apoio às UC: *i*) materiais produzidos no âmbito do DI (acetatos de apoio, livros dos regentes, etc.) que terão uma definição clara e um mapeamento de cobertura temática já conhecida; *ii*) e materiais de apoio externo (como livros, artigos, etc.) onde existe uma maior indefinição na sua cobertura temática.

- **Extensão de relações**, onde poderá ser mais "detalhado" um dos lados da relação. Útil quando se pretende manter a lógica de uma relação herdada, mas é necessário de refinar o âmbito do "alvo" ou "fonte" da relação (ou seja, indicar um subtipo específico em vez do conceito de mais alto nível).

5.1.1.3 Extensões futuras identificáveis

Já referido anteriormente, o conceito de material de referência (*di:referenceMaterial*) é uma situação que terá de ser revisitada e mais refinada. Deverá ser feito algum trabalho de análise dos diversos tipos de soluções já existentes para este problema, considerando-se a extensão do modelo para integração de uma solução mais robusta.

Refere-se igualmente a situação de revisão do BoK CC2001, consolidada no CC2008, que apesar de não introduzir alterações estruturais (foram apenas revisões do conteúdo), poderá fazer repensar a estrutura da secção CdC para a inclusão de sub-*curriculums*. A utilização experimental de uma eventual implementação do modelo poderá também levar ao refinar dos vários tipos de competências modeladas, descartando aquelas que tipicamente não se utilizarão. A zona das competências também é uma área onde se antevê a necessidade de remodelação no sentido de introduzir noções mais europeias, ou se possíveis globais, para fácil cruzamento inter-universidades.

5.1.2 Implementação do modelo e o protótipo

Os outros 2 aspectos a avaliar, além da extensibilidade, são a capacidade de modelação da informação originalmente pretendida e a capacidade de dar reposta às explorações originalmente propostas.

Relativamente ao primeiro aspecto referem-se alguns números obtidos do povoamento do modelo com informação, na fase inicial da construção do protótipo:

- Número de *AREAS*: 14;
- Número de *UNITS*: 138;
- Número de *TOPICS*: 815;
- Número de *LearningObjectives*: 663;
- Número de Skills (não LO): 96;
- Número de *Courses* (ACM-B): 44;

Estes números atestam a favor da capacidade de modelação do Corpo de Conhecimento proposto, podendo-se também dizer que, com base nas explorações propostas no protótipo (como as operações de descoberta de novas relações e o relatório que produz uma matriz das *units* nas *areas* dos vários *course*), esta informação foi correctamente modelada e poderá ser utilizada de futuro.

Relativamente à estruturação curricular, ou seja, os conceitos UC, Instância de UC, EC e OC, refere-se à partida que não se teve oportunidade de modelar o curso por inteiro, devido principalmente um único motivo: algum do detalhe pretendido no modelo que não existia totalmente descrito em documentos de suporte (como por exemplo, o "Proposta de curso de 1º ciclo de Licenciatura em Engenharia Informática" [71]) e requeria alguma validação junto dos docentes das disciplinas e respectivos responsáveis curriculares.

No entanto, conforme se pode verificar por algumas figuras da implementação do protótipo, foi possível modelar algumas UC (mostra-se o exemplo de "IP - Introdução à Programação") onde se efectuou um preenchimento dos vários aspectos (Units, *Skills, etc.*) com base em variada informação investigada e alguma intuição, sem que esta tivesse sido necessariamente validada por algum regente. Apesar de desta situação, que consideramos perfeitamente possível de solucionar com base em apoio do DI e dos seus regente, a modelação das peças base foram efectivamente possíveis apenas não se testando a mesma para várias situações (entenda-se várias disciplinas).

Relativamente às explorações, refere-se que aquilo que se demonstrou no protótipo, mais orientado à informação do CdC, é igualmente aplicável aos conceitos UC e EC, e caso existisse mais informação povoada nestes, era possível dar resposta às explorações propostas originalmente. Por exemplo, as seguintes explorações referidas, são casos praticamente semelhantes às apresentadas no protótipo.

- Cobertura temáticas de UCs
- Mapa temático detalhado do curso (seja tópicos, ou unidades ACM)
- Verificação de cruzamento entre temas abordados na UC;

Por fim, os aspectos de interface do protótipo demonstram que é importante ter especial atenção aos aspectos de apoio ao utilizador nos momentos de definição curricular. O conjunto de informação a preencher é extensa e considera-se importante ter mecanismos como, o preenchimento automático de informação relacionada, cálculos automatizados no preenchimento de tabelas numéricas, consulta e navegação apoiada de informação a seleccionar, algo que se demonstrou na implementação base da interface de UC.

5.2 Trabalho futuro

A proposta do modelo assume-se como um elemento base para um sistema a ser implementado de futuro. Este tipo de sistema poderá ser muito útil, conforme as explorações já referidas ao longo do documento, e é algo que poderá ser avaliado se é exequível desenvolver no DI.

Identificam-se alguns aspectos do modelo e seu povoamento, que se considera que terão de ser "refinados" antes de uma utilização mais séria:

- A indefinição relativa ao conceito referência bibliográfica e eventual povoamento com fontes conhecidas dentro do DI ou faculdade;
- Uma revisão do conceito competências, no sentido de verificar se é suficiente ou se é necessário estender, ou ainda simplificar a sua utilização no DI. Considera-se importante explorar no futuro possíveis fontes de competências mais *standard* e eventualmente internacionais, que poderão promover revisão deste aspecto;
- Considera-se importante efectuar uma investigação nos novos trabalhos ACM, "descendentes" do CC2005, no sentido de rever se a modelação se mantêm adequada (por exemplo, melhoras em relações como entre acmB:courseSyllabusTopic e acmA:Topic, que é actualmente inexistente) e também se é necessário repovoar os aspectos CdC (algo já detectado, devido as alterações CC2008).

Naturalmente, a implementação de um sistema "real", mais conforme à arquitectura proposta (que eventualmente poderá ter por base o protótipo), considera-se um aspecto futuro importante. Principalmente, os seguintes aspectos serão interessantes e relevantes:

- Capacidade de partilha de informação com o exterior;
- Alguns mecanismos de povoamento mais automatizado do modelo (eventualmente inspirados no trabalho do povoamento do CdC, executado para a construção inicial do protótipo);
- A disponibilização do sistema a vários tipos de utilizadores, principalmente a possibilidade de relatórios;

5.3 Conclusões

Considera-se que o principal objectivo da dissertação foi atingido. Ou seja, foi proposto nesta um modelo consistente com os principais objectivos. Isto é, a proposta de um modelo que:

- Tenha a sua aplicabilidade possível a vários tipos de cursos;
- Suporte informação sobre várias áreas de conhecimento;
- Considere aspectos de normalização e partilha da mesma;
- Seja possível representar de forma computável
- Tenha presente questões de extensibilidade;
- Permita de definição curricular das disciplinas, principalmente dos seus temas;
- Permita a organização e estrutura de planos curriculares do curso;
- Tenha presente considerações de Bolonha, relativas às competências;

Considera-se também que se atingiram parcialmente os objectivos secundários, a proposta de uma arquitectura modular e a implementação de um

protótipo exemplificativo. Refere-se parcialmente pois tinha sido a aspiração inicial deste autor em ter no protótipo um povoamento mais completo das UC do curso.

Assim, conclui-se que a definição e normalização de conceitos do ensino, aliado à recolha de informação sobre uma mesma base computável, permite produz efectivamente instrumentos úteis para a especificação, apoio à especificação e de análise sobre a completude, cobertura e qualidade dos cursos do ensino universitário.

Referências

1. *The Bologna Declaration of 19 June 1999*, *Joint declaration of the European Ministers of Education*. Disponível em: http://www.ond.vlaanderen.be/hogeronderwijs/bologna/documents/MDC/BOLOGNA_DECLARATION1.pdf
2. Declaração de Bolonha, Declaração conjunta dos ministros da educação europeus, assinada em Bolonha (19 de Junho de 1999). Disponível em: http://www.dges.mctes.pt/NR/rdonlyres/F9136466-2163-4BE3-AF08-C0C0FC1FF805/394/Declaracao_Bolonha_portugues.pdf
3. *The oficial Bologna Process website 2007-2010*. Disponível em: http://www.ond.vlaanderen.be/hogeronderwijs/bologna/
4. *Confederation of EU Rectors' Conferences and the Association of European Universities, The Bologna Declaration on the European space for higher education: an explanation*.
   Disponível em: http://ec.europa.eu/education/policies/educ/bologna/bologna.pdf
5. *The Bologna Process - Towards the European Higher Education Area.*
   Disponível em: http://ec.europa.eu/education/higher-education/doc1290_en.htm
6. *The framework of qualifications for the European Higher Education Area.* Disponível em: http://www.ond.vlaanderen.be/hogeronderwijs/bologna/documents/QF-EHEA-May2005.pdf
7. *European Credit Transfer and Accumulation System (ECTS)*. Disponível em: http://ec.europa.eu/education/lifelong-learning-policy/doc48_en.htm
8. *ECTS User's Guide*. Disponível em: http://ec.europa.eu/education/lifelong-learning-policy/doc/ects/guide_en.pdf
9. *European Credit Transfer and Accumulation System (ECTS), Key features*. Disponível em: http://ec.europa.eu/education/lifelong-learning-policy/doc/ects/key_en.pdf
10. Direcção Geral da Educação e Cultura, CARACTERÍSTICAS PRINCIPAIS DO ECTS.
    Disponível em: http://ec.europa.eu/education/lifelong-learning-policy/doc/ects/key_pt.pdf
11. Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (FCT-UNL), O Processo de Bolonha. Disponível em: http://www.fct.unl.pt/candidato/bolonha/
12. Departamento de Informática (D.I.) da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (FCT-UNL), Ensino no Departamento de Informática. Disponível em: http://di.fct.unl.pt/ensino/
13. Universidade Nova de Lisboa, O Processo de Bolonha, Linhas de Accção.
    Disponível em: http://www.unl.pt/bolonha/o-que-e-o-processo-de-bolonha/linhas-de-accao/linhas-de-accao
14. FCT – Estudante – http://www.fct.unl.pt/estudante/
15. D.I, FCT-UNL, O que são as Competências Complementares (Soft Skills)? Disponível em: http://di.fct.unl.pt/ensino/faq.php#q3

16. Litecky, C. R. Arnett, K. P. Prabhakar, B., *The paradox of soft skills versus technical skills in I.S. hiring, Journal of Computer Information Systems*, 2004, Vol 45; Part 1, pages 69-76.
17. Direcção Geral do Ensino Superior (DGES), Ministério da Ciência, Tecnologia e Ensino Superior (MCTES), O Processo de Bolonha. Disponível em: http://www.dges.mctes.pt/DGES/pt/Estudantes/Processo+de+Bolonha/Processo+de+Bolonha/
18. Direcção Geral do Ensino Superior (DGES), Ministério da Ciência, Tecnologia e Ensino Superior (MCTES), Situação em Portugal. Disponível em: http://www.dges.mctes.pt/DGES/pt/Estudantes/Processo+de+Bolonha/Processo+de+Bolonha/Situação+em+Portugal/
19. Assembleia da Republica, Lei n. 49/2005, de 30 de Agosto. Disponível em: http://www.sg.min-edu.pt/leis/lei_49_2005.pdf
20. Ministério da Ciência, Tecnologia e Ensino Superior (MCTES), Decreto-Lei nº 74/2006, de 24 de Março.
    Disponível em: http://www.mctes.pt/archive/doc/dl_2006_074.pdf
21. Ministério da Ciência, Tecnologia e Ensino Superior (MCTES), Decreto-Lei nº 107/2008, de 25 de Junho.
    Disponível em: http://www.mctes.pt/archive/doc/dl_2008_107.pdf
22. Ministério da Ciência, Tecnologia e Ensino Superior (MCTES), Gabinete do Ministro, Adequação do Ensino Superior ao Processo de Bolonha. Disponível em: http://www.portugal.gov.pt/pt/GC17/Governo/Ministerios/MCTES/Documentos/Pages/20070513_MCTES_Doc_Bolonha.aspx
23. Alfred V. Aho, Jeffrey D. Ullman, *Foundations of computer science*. 1992, *Computer Science Press*.
24. Peter J. Dennin, *Great principles in computing curricula*. *Proceedings of the 35th SIGCSE technical symposium on Computer science education*, 2004: p. 336-341.
25. *Association for Computing Machinery (ACM), ACM Computing Classification System TOC*. Disponível em: http://www.acm.org/about/class/
26. Coulter N., French J., Glinert E., Horton T., Mead N., Rada R., Ralston A., Rodkin C., Rous B., Tucker A., Wegner P., Weiss E., Wierzbicki C., *Association for Computing Machinery* (ACM), *Computing Classification System* 1998: *Current Status and Future Maintenance*. Disponível em: http://www.acm.org/about/class/ccsup.pdf
27. *Association for Computing Machinery (ACM), The 1998 ACM Computing Classification System*. Disponível em: http://www.acm.org/about/class/1998
28. *The Joint Task Force on Computing Curricula - IEEE Computer Society (IEEE-CS) & Association for Computing Machinery (ACM), Computing Curricula 2001 - Computer Science (Live SITE).* Disponível em: http://www.sigcse.org/resources/cs-2001
29. *The Joint Task Force on Computing Curricula - IEEE Computer Society (IEEE-CS) & Association for Computing Machinery (ACM), Computing Curricula 2001 - Computer Science, Final Report, December 15 2001 (PDF).* Disponível em: http://www.acm.org/education/curric_vols/cc2001.pdf

30. *The Joint Task Force on Computing Curricula - IEEE Computer Society (IEEE-CS) & Association for Computing Machinery (ACM), Computing Curricula 2001 - Computer Science, Appendix A - CS Body of Knowledge.*
Disponível em: http://www.sigcse.org/resources/cs-2001/appendixa
31. *The Joint Task Force for Computing Curricula 2005 - The Association for Computing Machinery (ACM) & The Association for Information Systems (AIS) & The Computer Society (IEEE-CS), Computing Curricula 2005 - The Overview Report, A volume of the Computing Curricula Series. 30 September 2005*. p:56. Disponível em: http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf
32. *American Mathematical Society (AMS), 2000 Mathematics Subject Classification*. Disponível em: http://www.ams.org/mathscinet/msc/msc.html
33. *American Mathematical Society (AMS), 2010 Mathematics Subject Classification*. Disponível em: http://www.ams.org/mathscinet/msc/msc2010.html
34. *The Internet Engineering Task Force (IETF), Network Working Group. RFC 3198 - Terminology for Policy-Based Management, Information Model*. Disponível em: http://tools.ietf.org/html/rfc3198
35. Veryard R*., Information modelling: practical guidance. Prentice Hall*, 1992. p:304
36. Mylopoulos J., *Information modeling in the time of the revolution*, (1998) *Information Systems*, 23 (3-4), pp. 127-155.
37. ACM-Association for Computing Machinery, Introduction to the 1998 ACM Computing Classification System, Valid through 2009. Disponível: http://www.acm.org/about/class/ccs98-intro
38. ACM-Association for Computing Machinery, The 1998 ACM Computing Classification System - List of Implicit Subject Descriptors in ACM CCS. Disponível em: http://portal.acm.org/lookup/ccsnoun.cfm
39. ACM-Association for Computing Machinery, The 1998 ACM Computing Classification System - HOW TO CLASSIFY WORKS USING ACM'S COMPUTING CLASSIFICATION SYSTEM. Disponível em: http://www.acm.org/about/class/how-to-use.
40. MIT. DSpace Federation. 2009. Disponível em: http://www.dspace.org/.
41. DSpace Dev @ Universidade do Minho: Desenvolvimento de ferramentas, extensões e add-ons para a plataforma DSpace. Disponível em: http://dspace-dev.dsi.uminho.pt:8080/pt/welcome.jsp
42. Dublin Core Metadata Initiative, 2009. Disponível em: http://dublincore.org/
43. Ferreira M., Baptista A.A., The use of taxonomies as a way to achieve interoperability and improved resource discovery in DSpace-based repositories. XATA 2005 : XML: aplicações e tecnologias associadas : actas da 3ª Conferência Nacional, Braga, 2005. ISBN 972-99166-1-6. p.67-79. Disponível em: http://xata.fe.up.pt/xata2005/papersfinal/25.pdf
44. DSpace Dev @ Universidade do Minho: ACM-CCS98. Disponível em: http://dspace-dev.dsi.uminho.pt:8080/pt/addon_acmccs98.jsp
45. Projects - Página CENTRIA da Dr.ª Susana Nascimento. Disponível em: http://centria.di.fct.unl.pt/~snt/projects.html

46. B. Mirkin, S. Nascimento, L. M. Pereira, (2007). ACM Classification Can Be Used for Representing Research Organizations, DIMACS Technical Report 2007-13, 18 p. Disponível em: http://centria.di.fct.unl.pt/~snt/papers/TR-13-2007-DIMACS.pdf
47. B. Mirkin, S. Nascimento, L. M. Pereira, (2008). Representing a Computer Science Research Organization on the ACM Computing Classification System, in P. Eklund, O. Haemmerl\'{e} (Eds.) Proc. 16th International Conference on Conceptual Structures, ISSN 1613-0073 (CEUR-WS.org), Toulouse, France, pp. 57-65. Disponível em: http://centria.di.fct.unl.pt/~snt/papers/ACMCL7_ICCS08.pdf
48. Allen B. Tucker, Bruce H. Barnes, Robert M. Aiken, Keith Barker, Kim B. Bruce, J. Thomas Cain, Susan E. Conry, Gerald L. Engel, Richard G. Epstein, Doris K. Lidtke, Michael C. Mulder, Jean B. Rogers, Eugene H. Spafford, and A. Joe Turner. Computing Curricula '91. Association for Computing Machinery and the Computer Society of the Institute of Electrical and Electronics Engineers, 1991.
49. John T. Gorgone, Gordon B. Davis, Joseph S. Valacich, Heikki Topi, David L. Feinstein, Herbert E. Longenecker Jr., Association for Computing Machinery (ACM) & Association for Information Systems (AIS) & Association of Information Technology Professionals (AITP), IS 2002 - Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. 2002. Disponível em: http://www.acm.org/education/education/curric_vols/is2002.pdf
50. The Joint Task Force on Computing Curricula - IEEE Computer Society (IEEE-CS) & Association for Computing Machinery (ACM), Software Engineering 2004 - Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, August 23, 2004. Disponível em: http://sites.computer.org/ccse/SE2004Volume.pdf
51. The Joint Task Force on Computing Curricula - IEEE Computer Society (IEEE-CS) & Association for Computing Machinery (ACM), Computer Engineering 2004 - Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering, 2004 December 12.
    Disponível em: http://www.acm.org/education/education/curric_vols/CE-Final-Report.pdf
52. Barry M. Lunt (Chair), Joseph J. Ekstrom, Sandra Gorka, Gregory Hislop, Reza Kamali, Eydie Lawson, Richard LeBlanc, Jacob Miller, Han Reichgelt. Association for Computing Machinery (ACM) & IEEE Computer Society (IEEE-CS), Information Technology 2008 - Curriculum Guidelines for Undergraduate Degree Programs in Information Technology, Nov. 2008.
    Disponível em: http://www.acm.org//education/curricula/IT2008%20Curriculum.pdf
    John T. Gorgone, ACM Co-Chair and Editor, Paul Gray, AIS Co-Chair and Editor, Edward A. Stohr, Joseph S. Valacich, Rolf T. Wigand, Communication of the Association for Information Systems (AIS), MSIS 2006: MODEL CURRICULUM AND GUIDELINES FOR GRADUATE DEGREE PROGRAMS IN INFORMATION SYSTEMS, June 2006.
    Disponível: http://www.acm.org/education/education/curric_vols/MSIS%202006.pdf
53. Association for Computing Machinery (ACM) & IEEE Computer Society (IEEE-CS), Computer Science Curriculum 2008: An Interim Revision of CS 2001 - Report from

the Interim Review Task Force, December 2008. Disponível em: http://www.acm.org//education/curricula/ComputerScience2008.pdf.

54. Integrated Software & Systems Engineering Curriculum (iSSEc) Project @ 2009 Stevens Institute of Technology - Graduate Software Engineering 2009(GSwE2009) - Curriculum Guidelines for Graduate Degree Programs in Software Engineering. Version 1.0, September 30, 2009. Disponível em: http://www.gswe2009.org/fileadmin/files/GSwE2009_Curriculum_Docs/GSwE2009_version_1.0.pdf
55. The Joint Task Force on Computing Curricula - IEEE Computer Society (IEEE-CS) & Association for Computing Machinery (ACM), Computing Curricula 2001 - Computer Science, Appendix B - Course Descriptions. Disponível em: http://www.sigcse.org/resources/cs-2001/appendixb
56. Pomerantz, J., Wildemuth, B. M., Yang, S., and Fox, E. A. 2006. Curriculum development for digital libraries. In Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (Chapel Hill, NC, USA, June 11 - 15, 2006). JCDL '06. ACM, New York, NY, 175-184.
57. Ronchetti, M. and Saini, P. 2004. Knowledge Management in an e-Learning System. In Proceedings of the IEEE international Conference on Advanced Learning Technologies (August 30 - September 01, 2004). ICALT. IEEE Computer Society, Washington, DC, 365-369.
58. Van Leeuwen, J. (Utrecht Univ., The Netherlands), Handbook of theoretical computer science (vol. A), MIT Press, Cambridge, MA, USA, 1991. p: 996.
59. Wilf, Herbert S. (University of Pennsylvania), Algorithms and Complexity, Internet Edition, Summer 1994. Disponível em: http://docs.happycoders.org/orgadoc/computer_science_theory/completeness_complexity/AlgComp.pdf
60. Saini P., Ronchetti M., 2003. Deriving ontology-based metadata for e-learning from the ACM computing curricula, # DIT-03-017 University of Trento – Department of Information and Communication Technology. Disponível em: http://eprints.biblio.unitn.it/archive/00000405/01/017.pdf
61. Ripamonti, L.A. and Peraboni, C. (2005). Hyperkrep Academic Communities: an ontology-based approach to content portability. Proc. of the International conference "Open Culture: accessing and sharing knowledge: scholarly production and education in the digital age", Milan, June 27-29 2005, Italy. Disponível em: http://www.aepic.it/conf/viewpaper.php?id=58&cf=3%20(HYPERKREP)
62. E. Duvak, W. Hodgins, S. Sutton, and S. L. Weibel, "Metadata principles and practicalities", D-Lib Magazine, vol. 8, no. 4, April 2002. [Online]. Disponível em: http://www.dlib.org/dlib/april02/weibel/04weibel.html
63. Extensible Markup Language (XML) - http://www.w3.org/XML/
64. The XML document type declaration - http://www.w3.org/TR/REC-xml/#dt-doctype
65. Horrocks. I. 2002 DAML+OIL: A description logic for the semantic web. IEEE Bulletin of the Technical Committee on Data Engineering, 25(1):4--9, 2002. Disponível em:

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.117.9373&rep=rep1&type=pdf#page=6

66. OWL Web Ontology Language Overview, W3C Recommendation 10 February 2004 - http://www.w3.org/TR/owl-features/
67. Berners-Lee T. , Hendler J. and Lassila O., 2001. The Semantic Web. Scientific American. May.
68. W3C Semantic Web Activity - http://www.w3.org/2001/sw/
69. Advanced Distributed Learning, 2004. Sharable Content Object Reference Model (SCORM) – The SCORM Content Aggregation Model, Version 1.3.
70. Departamento de Informática da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Proposta de curso de 1º ciclo de Licenciatura em Engenharia Informática - Adequacão do curso de Licenciatura em Engenharia Informática da FCT/UNL (publicada em D.R. II-Série Despacho no20 307/2004, de 29 de Setembro). Disponível em: http://www.di.fct.unl.pt/ensino/licenciatura/proposta_LEI.pdf
71. A report from a Joint Quality Initiative informal group - Shared 'Dublin' descriptors for Short Cycle, First Cycle, Second Cycle and Third Cycle Awards. Draft 1 working document on JQI meeting in Dublin on 18 October 2004. Disponível em: http://www.jointquality.nl/content/descriptors/CompletesetDublinDescriptors.doc
72. A.W.M. Meijers, C.W.A.M. van Overveld, J.C. Perrenet, Criteria for Academic Bachelor's and Master's Curricula. ISBN: 90-386-2217-1. Disponível em: http://www.jointquality.nl/content/descriptors/AC_English_Gweb.pdf
73. Bloom B. S. (1956). Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain. New York: David McKay Co Inc.
74. Department of Educational Psychology and Instructional Technology, University of Georgia. Emerging Perspectives on Learning, Teaching, and Technology - Bloom's Taxonomy . Disponível em: http://projects.coe.uga.edu/epltt/index.php?title=Bloom%27s_Taxonomy
75. Anderson, L. (2006, May). Revised Bloom's taxonomy. Paper presented at North Carolina Career and Technical Education Curriculum Development Training, Raleigh, NC.
76. L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, J. Raths, and M. C. Wittrock, A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Abridged Edition, 2nd ed. Allyn & Bacon, December 2000. Disponível em: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/080131903X e ainda em: http://sde.state.ok.us/Curriculum/CurriculumDiv/SocialStudies/pdf/Bloom's.ppt
77. Lillian N. Cassel, Russell L. Shackelford, Robert H. Sloan: A synthesis and ontology of all of computing. SIGCSE 2005: 65-66
78. Lillian N. Cassel, Robert H. Sloan, Gordon Davies, Heikki Topi, Andrew D. McGettrick: The computing ontology project: the computing education application. SIGCSE 2007: 519-520

79. The Ontology Project - Main WIKI. Disponível em: http://what.csc.villanova.edu/twiki/bin/view/Main/OntologyProject
80. Lillian N. Cassel, Gordon Davies, Richard LeBlanc, Lawrence Snyder, and Heikki Topi, Using a Computing Ontology as a Foundation for Curriculum Development. SWEL@ITS'08 - Accepted Papers. Disponível em: http://compsci.wssu.edu/iis/swel/SWEL08/Papers/Cassel.pdf
81. Meyer, B. 2006. Testable, Reusable Units of Cognition. Computer 39, 4 (Apr. 2006), 20-24. Disponível em: http://se.ethz.ch/~meyer/publications/computer/truc.pdf
82. Pedroni, M., Oriol, M., and Meyer, B. 2007. A framework for describing and comparing courses and curricula. SIGCSE Bull. 39, 3 (Jun. 2007), 131-135. Disponível em: http://www-users.cs.york.ac.uk/~manuel/Research/ITICSE2007/ITICSE2007.pdf
83. Eiffel Software's Open Source initiative. http://dev.eiffel.com/Main_Page
84. Standard ECMA-367 - Eiffel: Analysis, Design and Programming Language. 2nd edition (June 2006). Disponível em: http://www.ecma-international.org/publications/standards/Ecma-367.htm
85. Java homepage - http://www.java.com/
86. Trucstudio homepage - http://trucstudio.origo.ethz.ch/
87. Estrutura Curricular da Licenciatura em Engenharia Informática da FCT-UNL. Disponível em: http://www.di.fct.unl.pt/ensino/licenciatura-em-engenharia-informatica-1o-ciclo/estrutura-curricular
88. Plano de Estudos da Licenciatura em Engenharia Informática da FCT-UNL. Disponível em: http://www.di.fct.unl.pt/ensino/licenciatura-em-engenharia-informatica-1o-ciclo/plano-de-estudos
89. Kobayashi, M. and Takeda, K. 2000. Information retrieval on the web. ACM Comput. Surv. 32, 2 (Jun. 2000), 144-173.
90. Object Management Group, UNIFIED MODELING LANGUAGE - UML® Resource Page. Disponível em: http://uml.org/
91. Fowler, M. 2003 UML Distilled: a Brief Guide to the Standard Object Modeling Language. 3. Addison-Wesley Longman Publishing Co., Inc.
92. Namespaces in XML 1.0 (Second Edition). Disponível em: http://www.w3.org/TR/REC-xml-names/
93. Peckham, J. and Maryanski, F. 1988. Semantic data models. ACM Comput. Surv. 20, 3 (Sep. 1988), 153-189. Disponível em: http://ocw.kfupm.edu.sa/user062/ICS54101/p153-peckham-Semantic-Data-Models-1988.pdf
94. Chen, P. P. 1976. The entity-relationship model—toward a unified view of data. ACM Trans. Database Syst. 1, 1 (Mar. 1976), 9-36.
95. Stroustrup, Bjarne. The Design and Evolution of C++. Addison-Wesley. Reading, Mass. 1994. ISBN 0-201-54330-3. Page 49
96. Bibliographic Ontology Specification - Specification Document - 4 November 2009. Disponível em: http://bibliontology.com/specification

97. Erasmus Programme - Students Mobility for Study (SMS) - http://www.fct.unl.pt/student/mobility/sms
98. Eckerson, Wayne W. "Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications." Open Information Systems 10, 1 (January 1995): 3(20).
99. Alonso, G., Casati, F., Kuno, H., Machiraju, V. 2004. Web services: concepts, architectures and applications, 2004, 354 p., Hardcover, ISBN: 978-3-540-44008-6
100. Ralph Kimball and Margy Ross. 2002. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (2nd ed.). John Wiley & Sons, Inc., New York, NY, USA.
101. Informatics Europe - Informatics Europe is the association of computer science departments of universities and research laboratories, public and private, in Europe and neighbouring areas - http://www.informatics-europe.org/
102. McKay, David J.C. (2003). Information Theory, Inference, and Learning Algorithms. Cambridge University Press. ISBN 0-521-64298-1.
103. Rish, Irina. (2001). "An empirical study of the naive Bayes classifier". IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence. Disponível em: http://www.research.ibm.com/people/r/rish/papers/RC22230.pdf
104. Deerwester, S., et al, Improving Information Retrieval with Latent Semantic Indexing, Proceedings of the 51st Annual Meeting of the American Society for Information Science 25, 1988, pp. 36–40.
105. JSON (JavaScript Object Notation) - http://www.json.org/
106. D. Heinemeier Hansson, World of Resources, RailsConf 2006 Keynote Presentation.
107. Robert Battle and Edward Benson. 2008. Bridging the semantic Web and Web 2.0 with Representational State Transfer (REST). Web Semant. 6, 1 (February 2008), 61-69.
108. N. Viana, R. Raminhos, and J. M. Pires, "A real time data extraction, transformation and loading solution for semi- structured text files," in EPIA, ser. Lecture Notes in Computer Science, C. Bento, A. Cardoso, and G. Dias, Eds., vol. 3808. Springer, 2005, pp. 383–394.
109. Kobayashi, M. and Takeda, K. (2000). "Information retrieval on the web". ACM Computing Surveys (ACM Press) 32 (2): 144–173.
110. Konchady Manu "Text Mining Application Programming (Programming Series)" by Manu Konchady, Charles River Media, ISBN 1584504609.
111. Ikonomakis, M., Kotsiantis, S. Tampakas, V., Text Classification Using Machine Learning Techniques, WSEAS Transactions on Computers, Issue 8, Volume 4, August 2005, pp. 966-974. Disponível em: http://www.math.upatras.gr/~esdlab/en/members/kotsiantis/Text%20Classification%20final%20journal.pdf
112. Resource Description Framework (RDF). 2004; Disponível em: http://www.w3.org/RDF/
113. URIs, URLs, and URNs: Clarifications and Recommendations 1.0; Disponível em: http://www.w3.org/TR/uri-clarification/

114. RDF/XML Syntax Specification (Revised). Disponível em: http://www.w3.org/TR/REC-rdf-syntax/#rdfxml
115. Notation 3 Specification on W3C Design Issues by Tim Berners-Lee. 1998; Disponível em: http://www.w3.org/DesignIssues/Notation3.html
116. Notation 3, Wikipedia; Disponível em: http://en.wikipedia.org/wiki/Notation3
117. RDF Vocabulary Description Language 1.0: RDF Schema. 2004. Disponível em: http://www.w3.org/TR/rdf-schema/
118. The Protégé Ontology Editor and Knowledge Acquisition System. 2006; Disponível em: http://protege.stanford.edu/.
119. S. Melnik, Storing RDF in a relational database. Disponível em: http://infolab.stanford.edu/~melnik/rdf/db.html
120. R. Agrawal, A. Somani, and Y. Xu: Storage and Querying of E-Commerce Data, Proc. VLDB 2001, Roma, Italy; Disponível em: http://www.vldb.org/conf/2001/P149.pdf
121. Lehigh University Triplestore Benchmark. Disponível em: http://swat.cse.lehigh.edu/projects/lubm/
122. SPARQL Query Language for RDF. W3C Recommendation 15 January 2008; Disponível em: http://www.w3.org/TR/rdf-sparql-query/
123. A Semantic Web Framework for Java - http://jena.sourceforge.net/
124. Redland RDF Libraries - http://librdf.org/
125. OpenRDF, Sesame - http://www.openrdf.org/
126. RubyOnRails - http://rubyonrails.org/
127. David Heinemeier Hansson - http://www.loudthinking.com
128. 37signals - http://37signals.com/
129. Ruby Programming Language - http://www.ruby-lang.org/
130. Fielding, Roy T.; Taylor, Richard N. (2002-05), "Principled Design of the Modern Web Architecture", ACM Transactions on Internet Technology (TOIT) (New York: Association for Computing Machinery) 2 (2): 115–150. Disponível em: http://www.ics.uci.edu/~taylor/documents/2002-REST-TOIT.pdf
131. ActiveRDF - "putting the semantic web on rails" - http://activerdf.org/
132. Eyal Oren, Renaud Delbru, Sebastian Gerke, Armin Haller, and Stefan Decker. 2007. ActiveRDF: object-oriented semantic web programming. In Proceedings of the 16th international conference on World Wide Web (WWW '07). ACM, New York, NY, USA, 817-824. Disponível em: http://www.eyaloren.org/pubs/www2007.pdf
133. SQLite - http://www.sqlite.org/
134. Incremental Language Description of OWL DL and OWL Full - http://www.w3.org/TR/2004/REC-owl-features-20040210/#s4
135. HTML Tidy Library Project - http://tidy.sourceforge.net/
136. XSL Transformations (XSLT), Version 1.0. W3C Recommendation 16 November 1999 - http://www.w3.org/TR/xslt