

CLAV - Modelação e Especificação do modelo formal

Mestrado Integrado em Engenharia Informática - Laboratório em Engenharia Informática 18/19

Autores:

- Armando Santos
- Gonçalo Duarte
- Prof. José Carlos Ramalho
- Prof. José Nuno Oliveira

Alloy

- O modelo Alloy encontra-se em `Alloy/clav.als` .
- O tema para visualizar instancias em Alloy é o `Alloy/theme2.thm`

Índice

- [CLAV - Modelação e Especificação do modelo formal](#)
- [Alloy](#)
- [Índice](#)
- [Lista consolidada](#)
 - [Níveis](#)
 - [Classes de nível 3](#)
 - [Classes de nível 4](#)
- [Classes](#)
- [Relações de cada Classe](#)
 - [ReferencialClassificativo](#)
 - [ReferencialClassificativo](#)
 - [Classe](#)
 - [Classe_N1](#)
 - [Classe_N2](#)
 - [Classe_N3](#)

- Classe_N4
- Legislacao
- EntidadeResponsavel
- Entidade
- TipologiaEntidade
- TermoIndice
- CriterioJustificacao
- DestinoFinal
- Justificacao
- NotaAplicacao
- NotaExclusao
- PCA
- Invariantes
 - Indice
 - Classes de 1º Nível
 - Classes de 2º Nível
 - Classes de 3º Nível com desdobramento.
 - Invariantes sobre os PNs (Classe 3)
 - Invariantes sobre a relação Suplementar: implicações no PCA
 - Invariantes sobre a relação Síntese: implicações no DF
 - Invariantes sobre a relação Complementar: implicações no DF
 - Invariantes sobre o DF
 - Invariantes sobre o PCA
 - Invariantes sobre os Termos de Índice
 - Invariantes das relações `temRelProc`
 - Invariantes sobre as Justificações
- Modelação
 - Relações inversas
 - Visualização
 - Problemas

Lista consolidada

É uma estrutura **hierárquica de classes** criada para a classificação dos processos de negócio (PN's) da AP, **constituída por 4 níveis**.

Níveis

Todas as classes e processos são constituídos pelos seguintes atributos:

- Código;
- Título;
- Descrição;
- Notas de Aplicação (**Exceto classes de nível 4**);
- Notas de Exclusão (**Exceto classes de nível 4**).

Classes de nível 3

As classes de nível 3 são relativas a **processos de negócio (PN)** e têm mais 2 atributos:

- Tipo de Processo;
- Processo Transversal.

Também possuem os 4 atributos que pertencem aos de classe nível 4.

Classes de nível 4

- Prazo de conservação administrativa;
- Justificação do prazo de conservação;
- Destino Final;
- Justificação do destino final.

Classes

Classes inferidas a partir das interrogações que a ontologia deve responder:

- Lista Consolidada
- Classe de N1
- Classe de N2
- Classe de N3
- Classe de N4
- Entidade
- Legislação
- Tabela de Seleção

- Tipologia
- Utilizador

Neste projeto estamos interessados apenas na Lista Consolidada. Sendo assim, refinamos as classes relevantes, filtrando as que não têm interesse para o modelo do problema a tratar e adicionando outras justificadas pela observação da ontologia já definida. Após várias iterações chegamos à lista final de classes a modelar:

- ReferencialClassificativo
- ListaConsolidada
- TabelaSelecao
- Classe
- Classe de N1
- Classe de N2
- Classe de N3
- Classe de N4
- EntidadeResponsavel
- Entidade
- TipologiaEntidade
- Legislacao
- RelacaoPesada
- TermoIndice
- CriterioJustificacao
- CriterioJustificacaoComplementaridadeInfo
- CriterioJustificacaoDensidadeInfo
- CriterioJustificacaoGestionario
- CriterioJustificacaoLegal
- CriterioJustificacaoUtilidadeAdministrativa
- DestinoFinal
- ExemploNotaAplicacao
- Justificacao
- NotaAplicacao
- NotaExclusao
- PCA

As classes no segundo nível de indentação têm uma relação 'subClassOf' associada inerentemente.

Relações de cada Classe

No documento que possuí a modelação e análise da ontologia não estão documentadas muitas das relações, sendo assim, nesta secção pretendemos reunir e documentar todas as relações existentes na ontologia.

Após várias sessões de trabalho e reuniões com os Professores chegamos ao que pensamos ser o apanhado final que se segue a seguir.

ReferencialClassificativo

- `temClasse :: ReferencialClassificativo -> Classe`

ReferencialClassificativo

- ``temClasse :: ReferClasse`

Classe

Classe Abstrata (Classe_N1, Classe_N2, Classe_N3, Classe_N4)

- `pertenceLC :: Classe -> ListaConsolidada`
- `pertenceTS :: Classe -> TabelaSelecao`
- `temNotaAplicacao :: Classe -> NotaAplicacao` (**Exceto Classe_N4**)
- `temNotaExclusao :: Classe -> NotaExclusao` (**Exceto Classe_N4**)
- `temExemploNA :: Classe -> ExemploNotaAplicacao` (**Exceto Classe_N4**)

Classe_N1

- `temFilho :: Classe_N1 -> Classe_N2`

Classe_N2

- `temPai :: Classe_N2 -> Classe_N1`

- `temFilho :: Classe_N2 -> Classe_N3`

Classe_N3

- `temPai :: Classe_N3 -> Classe_N2`
- `temDono :: Classe_N3 -> Entidade+Tip`
- `temFilho :: Classe_N3 -> Classe_N4`
- `pertenceTS :: Classe_N3 -> TabelaSelecao`
- `temTI :: Classe_N3 -> TermoIndice`
- `temRelProc`
- `eAntecessorDe :: Classe_N3 -> Classe_N3`
- `eComplementarDe :: Classe_N3 -> Classe_N3`
- `eCruzadoCom :: Classe_N3 -> Classe_N3`
- `eSinteseDe :: Classe_N3 -> Classe_N3`
- `eSintetizadoPor :: Classe_N3 -> Classe_N3`
- `eSucessorDe :: Classe_N3 -> Classe_N3`
- `eSuplementoDe :: Classe_N3 -> Classe_N3`
- `eSuplementoPara :: Classe_N3 -> Classe_N3`
- `temParticipante`
- `temParticipanteComunicador :: Classe_N3 -> Tip+Ent`
- `temParticipanteIniciador :: Classe_N3 -> Tip+Ent`
- `temParticipanteApreciador :: Classe_N3 -> Tip+Ent`
- `temParticipanteDecisor :: Classe_N3 -> Tip+Ent`
- `temParticipanteAssessor :: Classe_N3 -> Tip+Ent`
- `temParticipanteExecutor :: Classe_N3 -> Tip+Ent`
- `temLegislacao :: Classe_N3 -> Legislacao`
- `temDF :: Classe_N3 -> DestinoFinal`
- `temPCA :: Classe_N3 -> PCA`

Classe_N4

- `temPai :: Classe_N4 -> Classe_N3`
- `temTI :: Classe_N4 -> TermoIndice`
- `temRelProc`
- `eSinteseDe :: Classe_N4 -> Classe_N4`
- `eSintetizadoPor :: Classe_N4 -> Classe_N4`

- `temDF :: Classe_N4 -> DestinoFinal`
- `temPCA :: Classe_N4 -> PCA`

Legislacao

- `temEntidadeResponsavel :: Legislacao -> Entidade+Tip`

EntidadeResponsavel

- `eDonoProcesso :: EntidadeResponsavel -> Classe_N3`
- `participaEm`
- `participaEmComunicando :: EntidadeResponsavel -> Classe_N3`
- `participaEmIniciando :: EntidadeResponsavel -> Classe_N3`
- `participaEmApreciando :: EntidadeResponsavel -> Classe_N3`
- `participaEmDecidindo :: EntidadeResponsavel -> Classe_N3`
- `participaEmAssessorando :: EntidadeResponsavel -> Classe_N3`
- `participaEmExecutando :: EntidadeResponsavel -> Classe_N3`

Entidade

- `pertenceTipologiaEnt :: Entidade -> Tipologia`

TipologiaEntidade

- `contemEntidade :: TipologiaEntidade -> Entidade`

TermoIndice

- `estaAssocClasse :: TermoIndice -> Classe_N3 + Classe_N4`

CriterioJustificacao

- `critTemLegAssoc :: CriterioJustificacao -> Legislacao`
- `critTemProcRel :: CriterioJustificacao -> Classe_N3`

DestinoFinal

- `temJustificacao :: DF -> Justificacao`

Justificacao

- `temCritério :: Justificacao -> CritérioJustificacao`

NotaAplicacao

- `naPertenceClasse :: NotaAplicação -> Classe_N1 + Classe_N2 + Classe_N3`

NotaExclusao

- `nePertenceClasse :: NotaExclusao -> Classe_N1 + Classe_N2 + Classe_N3`
- `usarClasse :: NotaExclusao -> Classe`

PCA

- `temJustificacao :: PCA -> Justificacao`

Invariantes

Foi analisado o documento com os requisitos funcionais e invariantes a preservar sobre o problema. Mais uma vez a documentação existente não estava nas melhores condições e surgiu a necessidade de organizar e documentar cada um dos invariantes.

Tendo isto em conta, esta secção organiza todos os invariantes até agora especificados sendo que podem vir a ser adicionados/removidos alguns no futuro caso surjam alterações imprevistas ou caso a sua coexistência não seja consistente.

Existem pequenas anotações ao longo de certos invariantes. Estas anotações correspondem a certas dúvidas que foram surgindo relativas ao invariante em questão e respondidas pelo Professor em reunião. Sendo assim, estas anotações mantiveram-se no documento para auxiliar a leitura dos invariantes e possivelmente esclarecer o leitor. Ainda em relação às anotações; anotações do tipo **Check** indicam que o invariante já se encontra especificado em Alloy; anotações do tipo **NOVO** representam invariantes que foram adicionados durante o trabalho e que não constavam na especificação inicial.

A numeração dos invariantes corresponde à numeração do Alloy.

Indice

- [inv1](#)
- [inv2](#)
- [inv3](#)
- [inv4](#)
- [inv5](#)
- [inv6](#)
- [inv7](#)
- [inv8](#)
- [inv9](#)
- [inv10](#)
- [inv11](#)
- [inv12](#)
- [inv13](#)
- [inv14](#)
- [inv15](#)
- [inv16](#)
- [inv17](#)
- [inv18](#)
- [inv19](#)
- [inv20](#)
- [inv21](#)
- [inv22](#)
- [inv23](#)
- [inv24](#)
 - [inv24-1](#)
 - [inv24-2](#)
 - [inv24-3](#)
- [inv25](#)
- [inv26](#)
- [inv27](#)
- [inv28](#)

- [inv29](#)
- [inv30](#)
- [inv31](#)
- [inv32](#)
- [inv33](#)
- [inv34](#)
- [inv35](#)
- [inv36](#)
- [inv37](#)
- [inv38](#)

Classes de 1º Nível

- **(inv1)** Se uma Classe_N1 pertence a uma LC/TS, consequentemente os seus filhos, netos, etc.. também têm de pertencer. **(NOVO) (Check)**
- **(inv2)** Se uma Classe_N1 não pertence a uma LC/TS, consequentemente os seus filhos, netos, etc.. também não pertencem. **(NOVO) (Check)**
- **(inv24.1)** 2 Classe_N1 não podem ter a mesma instância NotaAplicacao **(NOVO) (Check)**
- **(23)** 2 Classe_N1 não podem ter a mesma instância NotaExclusao **(NOVO) (Check)**
- **(23)** 2 Classe_N1 não podem ter a mesma instância ExemploNotaAplicacao **(NOVO) (Check)**

Classes de 2º Nível

- **(inv24.2)** 2 Classe_N3 não podem ter a mesma instância NotaAplicacao **(NOVO) (Check)**
- **(24)** 2 Classe_N3 não podem ter a mesma instância NotaExclusao **(NOVO) (Check)**

- (24) Classe_N3 não podem ter a mesma instância
ExemploNotaAplicacao (**NOVO**) (**Check**)

Classes de 3º Nível com desdobramento.

- Classes_N3 com desdobramento:
 - **(inv5)** Os 4ºs níveis herdam as legislações existentes no 3º nível (subconjunto), quer para o PCA quer para o DF (**Check**)
 - **(inv6)** Só existe desdobramento caso o PCA ou DF sejam distintos (**NOVO**) (**Check**)
 - PCA distinto
 - deve ser possível diferenciar as relações de cada uma das subdivisões ao 4º nível (por exemplo o critério de utilidade administrativa em cada um dos 4ºs níveis reflete relações suplementares com 3ºs níveis distintos; ou apenas uma subdivisão de 4º nível tem critério legal) **??? Vão ter um critério de utilidade administrativa que referenciam classes nível 3 distintas entre eles.**
(**PENDENTE**)
 - **(inv7)** O DF é avaliado tendo em conta o contexto de 3º nível, em que:
 - Se for distinto tem que haver uma relação de síntese entre as classes 4 filhas. (**Check**)
 - Se for igual não interessa (**Check**)
 - **(inv8)** DF distinto
 - Deve haver uma relação de síntese (de ou por) entre as classes 4 filhas (**Check**)
 - O PCA é igual (**Check**)
 - **(inv9)** Os termos de índice vão para os 4ºs níveis (repetidos) (**Check**)
- **(inv3)** As relações `temDF` e `temPCA`, não existem numa classe 3 se esta tiver filhos. (**Check**)
- **(inv4)** As relações `temDF` e `temPCA`, existem numa classe 3 se esta **não** tiver filhos. (**NOVO**) (**Check**)
- Um PN tem sempre um PCA e um DF. O valor do PCA poderá ser nulo se o PCA tiver uma nota associada. (**Check**)

- **(inv10)** Se um PN (Classe 3) for **complementar** de outro que se desdobra ao 4º nível, é necessário, com base no critério de complementaridade informacional, a relação manter-se ao 3º nível. Pelo menos um dos 4ºs níveis deve ser de **conservação**. **(Check)**
- **(inv11)** Um processo só tem participantes se for **transversal** (O campo 'transversal' tiver o valor sim). **(Check)**
- Uma classe N3 tem de ter sempre pelo menos um dono. **(Check)**
- **(inv24.3)** 2 Classe_N3 não podem ter a mesma instância NotaAplicacao **(NOVO)** **(Check)**
- (25) 2 Classe_N3 não podem ter a mesma instância NotaExclusao **(NOVO)** **(Check)**
- (25) 2 Classe_N3 não podem ter a mesma instância ExemploNotaAplicacao **(NOVO)** **(Check)**
- **(inv25)** 2 Classe_N3 não podem ter os mesmos filhos **(NOVO)** **(Check)**
- **(inv26)** 2 Classe_N3 não podem ter o mesmo TI **(NOVO)** **(Check)**

Invariantes sobre os PNs (Classe 3)

- **(inv12)** As relações `eComplementarDe` e `eCruzadoCom` são simétricas. **(Check)**
- **(inv14)** As relações `eSinteseDe`, `eSucessorDe` e `eSuplementoDe` são antisimétricas. (`eSuplementoDe` ou `eSuplementoPara` ?? As duas) **(Check)**
- **(inv13)** Na relação `temRelProc` um PN não se relaciona com ele próprio. **(NOVO)** **(Check)**
- **(inv15)** Um PN só pode ter uma relação com outro PN **(Uma relação como assim?? R: Um PN não pode ter relações distintas com o mesmo processo | Que relações?? As do tipo Classe_N4 -> Classe_N3)** **(Check)**
- **(inv16)** Se um PN se desdobra em 4ºs níveis os Termos de Índice passam para os filhos. **(Check)**
- **(inv17)** Os termos de índice de um PN não existem em mais nenhuma classe 3 **(NOVO)** **(Check)**
- **(inv20)** Dois PNs não podem ter a mesma instância de DF. **(NOVO)** **(Check)**
- **(inv21)** Dois PNs não podem ter a mesma instância de PCA. **(NOVO)** **(Check)**

Invariantes sobre a relação Suplementar: implicações no PCA

- **(inv29)** Quando o PN (sem filhos) em causa `eSuplementoPara` outro, deve ser acrescentado um critério de utilidade administrativa na justificação do respetivo PCA; (`eSuplementoDe` **OU** `eSuplementoPara` ?? **SuplementoPara** | **Que PCA, o do PN em causa ou o outro? Em causa** | **Não existe nenhuma relação do tipo** `Justificacao` -> `CriterioJustificacao` **R: temCriterio**) (**Check**)
 - Nesse critério, critério de utilidade administrativa, devem aparecer todos os processos com os quais existe uma relação `eSuplementoPara` ; (**Que todos os processos?? Todas as que o PN em CAUSA está relacionado por a relação** `eSuplementoPara`)
- **(inv33)** Quando o PN (com filhos) em causa `eSuplementoPara` outro, deve ser acrescentado um critério de utilidade administrativa na justificação PCA dos filhos em que os processos relacionados com o critério podem ser distintos. (**NOVO**) (**Check**)

Invariantes sobre a relação Síntese: implicações no DF

- Quando o PN em causa é síntese de outro, o DF deve ter o valor de “Conservação” (**Check**);
- Quando o PN em causa é sintetizado por outro, o DF deve ter o valor de “Eliminação” (**Check**);
- **(inv34)** Todos os processos relacionados (sem filhos) por uma relação de síntese deverão estar relacionados com o critério de densidade informacional da respetiva justificação. (`critTemProcRel` **O critério é que se relaciona com os processos**) (**Check**)
- **(inv35)** Todos os processos relacionados (com filhos) por uma relação de síntese, os filhos deverão estar relacionados com o critério de

densidade informacional da respetiva justificação. (critTemProcRel O
critério é que se relaciona com os processos) (NOVO) (Check)

- ~~Para a relação eSintetizadoPor se houver uma relação complementar isto não se aplica~~ (???)

Invariantes sobre a relação Complementar: implicações no DF

- Uma relação de complementaridade implica a conservação dos PNs que mantêm essa relação; (implica que o DF seja de conservação? Nos dois processos? Sim e sim) (Check)
- (inv36) Todos os processos relacionados (sem filhos) pela relação complementarDe , devem estar relacionados com o critério de complementaridade informacional da respetiva justificação. (Check)
- (inv37) Todos os processos relacionados (com filhos) pela relação complementarDe , os filhos devem estar relacionados com o critério de complementaridade informacional da respetiva justificação. (NOVO) (Check)

Invariantes sobre o DF

- (inv30) Um DF, na sua justificação, deverá conter apenas critérios de densidade informacional, complementaridade informacional e legal (Check)
- (inv18) 2 DFs não podem ter a mesma instancia de Justificacao; (NOVO) (Check)
- (inv27) Um DF, na sua justificação, apenas deve ter uma instancia de JustificacaoDF; (NOVO) (Check)

Invariantes sobre o PCA

- (inv31) Um PCA, na sua justificação, deverá conter apenas critérios gestores, utilidade administrativa e legal (Check)
- (inv19) 2 PCAs não podem ter a mesma instancia de Justificacao; (NOVO) (Check)

- **(inv28)** Um PCA, na sua justificação, apenas deve ter uma instancia de JustificacaoPCA; **(NOVO)** **(Check)**

Invariantes sobre os Termos de Índice

- Os termos de índice são únicos no universo do sistema **(Check)**

Invariantes das relações `temRelProc`

Só se aplica ao 3º nível.

- **(inv32)** Se um PN é (por ordem de prioridade): **(NOT CHECK)**
 - `eComplementarDe` -> DF é de conservação
 - `eSinteseDe` -> DF é de conservação
 - `eSintetizadoPor` -> DF é de eliminação
 - nenhuma das acima -> DF é NE (Não especificado)

Invariantes sobre as Justificações

- **(inv23)** 2 Justificações nao podem ter a mesma instancia de `CriterioJustificacao` **(NOVO)** **(Check)**
- **(inv22)** Cada Justificacao tem no maximo 3 `CriterioJustificacao` diferentes **(NOVO)** **(Check)**
- **(inv38)** Cada Justificacao tem no maximo 1 Criterio de cada tipo **(NOVO)** **(Check)**

Modelação

Após ter sido feito o apanhado de todas as classes e relações entre elas irá ser iniciada a modelação do domínio do problema. Durante a especificação dos invariantes chegamos à conclusão de que será necessário ter em conta algumas relações de atributos como a `dfValor :: DF -> String` e a `processoTransversal :: Classe_N3 -> String`. Refinando estas últimas relações podemos simplificar a `processoTransversal :: Classe_N3 -> Boolean`.

A modelação irá ser feita utilizando o Alloy Analyzer que fornece uma linguagem de modelação e descrição de estruturas e uma ferramenta para as explorar. Um modelo Alloy é um conjunto de restrições que descrevem (implicitamente) um conjunto de estruturas. A ferramenta Alloy Analyzer, é

um *solver* que pega nas restrições do modelo e encontra estruturas que as satisfaçam e permite verificar propriedades de um modelo gerando contra-exemplos.

Após ter transportado o domínio do problema, traduzindo em *signatures* cada classe, para o Alloy, raciocínamos acerca da cardinalidade de certas classes assim como a taxonomia das relações (endo-relações).

Relações inversas

Todas as relações possuem uma inversa (relação inversa implícita), no entanto na ontologia é necessário especifica-las manualmente. Devido a isto nem todas as relações inversas estão especificadas na ontologia. Em Alloy existe a noção implícita de relação inversa sendo que não seria necessário especificá-las no modelo do problema mas, uma vez que a nomeação de cada par (relação/relação inversa) em alguns casos não é intuitivo decidimos incluí-las no modelo, associando a cada uma as seguintes restrições (factos):

- `pertenceLC° = temClasse`
- `temFilho° = temPai`
- `temNotaAplicacao° = naPertenceClasse`
- `temNotaExclusao° = nePertenceClasse`
- `eDonoProcesso° = temDono`
- `temTI° = estaAssocClasse`
- `eAntecessorDe° = eSucessorDe`
- `eSinteseDe° = eSintetizadoPor`
- `eSuplementoDe° = eSuplementoPara`
- `temParticipanteComunicador° = participaEmComunicando`
- `temParticipanteIniciador° = participaEmIniciando`
- `temParticipanteApreciador° = participaEmApreciando`
- `temParticipanteAssessor° = participaEmAssessorando`
- `temParticipanteExecutor° = participaEmExecutando`
- `temLegislacao° = VER NO FUTURO`
- `contemEntidade° = pertenceTipologiaEnt`

Uma ideia futura será a de abstrair do modelo a definição explícita de relações inversas e, com o auxílio de um migrador/processador converter a especificação do modelo, numa especificação em OWL.

Visualização

Como já foi mencionado o Alloy Analyzer permite visualizar instâncias, por ele geradas, que satisfazem as restrições especificadas. No entanto devido à complexidade do problema em mãos será preciso ter a capacidade de visualizar estas instâncias de forma perceptível. É possível configurar um tema de visualização que se adeque e permita raciocinar de um modo geral sobre o que está a acontecer dentro do modelo.

O primeiro contacto com o modelo especificado (apenas os seus objetos e relações entre eles) foi a seguinte:

O modelo irá simplificar à medida que formos especificando os invariantes e configurando o tema de visualização.

Uma primeira tentativa de configuração do tema de visualização, para uma instância que mais se aproxima de um caso real:

Problemas