

5

Meta-Modelo de Execução de Consultas

O *framework* QEEF deve ser instanciado de acordo com os seus pontos de adaptação, vistos no capítulo anterior, produzindo uma MEC específica para suportar diferentes modelos de dados. Para suportar um modelo de execução, a deve-se instanciar um conjunto de módulos que são combinados num PEC tradicional para que este seja executado segundo um modelo de execução desejado. Neste capítulo, é descrito um meta-modelo de execução de consultas denominado de QUEM (*QUery Execution Meta-model*) para permitir a combinação de módulos dentro de um PEC submetido pelo usuário.

Tradicionalmente, uma MEC é construída para suportar um modelo de execução de consultas específico. Neste contexto, os PECs submetidos à MEC contêm operadores algébricos (plano algébrico) e de controle e são executados sempre segundo um único modelo de execução. Numa MEC instanciada pelo QEEF, os PECs podem incluir, além do plano algébrico, módulos de execução, como veremos a seguir.

5.1.

Meta-Modelo QUEM

A especificação dos PECs no QEEF pode ser feita de duas formas:

- de forma dependente da MEC instanciada
- de forma independente da MEC instanciada

No primeiro caso, o usuário especifica os PECs utilizando operadores algébricos e de controle como é feito nas MECs tradicionais. Neste caso, o usuário torna-se mais dependente da implementação da MEC à medida que ele não se beneficia dos módulos existentes na MEC nem utiliza o meta-modelo QUEM. No segundo caso, o usuário especifica os PECs de forma mais uniforme utilizando apenas operadores algébricos (do plano algébrico) e os módulos de execução de

acordo com o meta-modelo QUEM, que define as regras de combinação entre operadores algébricos e módulos (existentes ou que possam surgir).

Os PECs especificados segundo o QUEM são denominados de meta-PECs e são pré-processados pelo QEEF e, portanto, independem da MEC instanciada, resultando em PECs finais que, por sua vez, são executados pela MEC instanciada gerando o resultado da consulta como mostra a Figura 39.

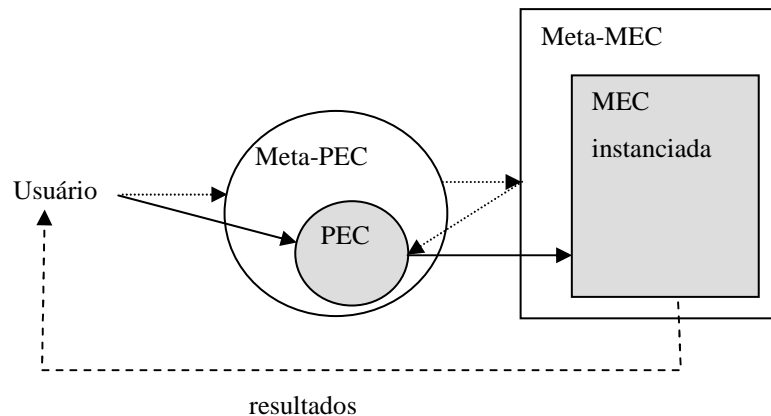


Figura 39 - Execução de Consultas no QEEF

Na etapa de pré-processamento do meta-PEC, cada módulo especificado é executado pelo QEEF resultando na adição dos operadores de controle apropriados entre os operadores algébricos, segundo a lógica de implementação de cada módulo. Como resultado, teremos um PEC final composto apenas por operadores algébricos e de controle e seus relacionamentos produtor-consumidor.

Na etapa de execução do PEC final, a MEC instanciada executará os operadores algébricos e de controle gerando o resultado da consulta para a aplicação (usuário).

As duas etapas descritas acima estão associadas aos dois níveis de execução do QEEF, ilustrados na Figura 40.

- No primeiro nível, o QEEF se comporta como uma meta-Máquina (máquina virtual), executando os módulos m_1, m_2, \dots, m_N (operadores virtuais) contidos num meta-PEC (plano virtual).

- No segundo nível, o QEEF se comporta como uma Máquina instanciada executando os operadores algébricos e de controle contidos no PEC.

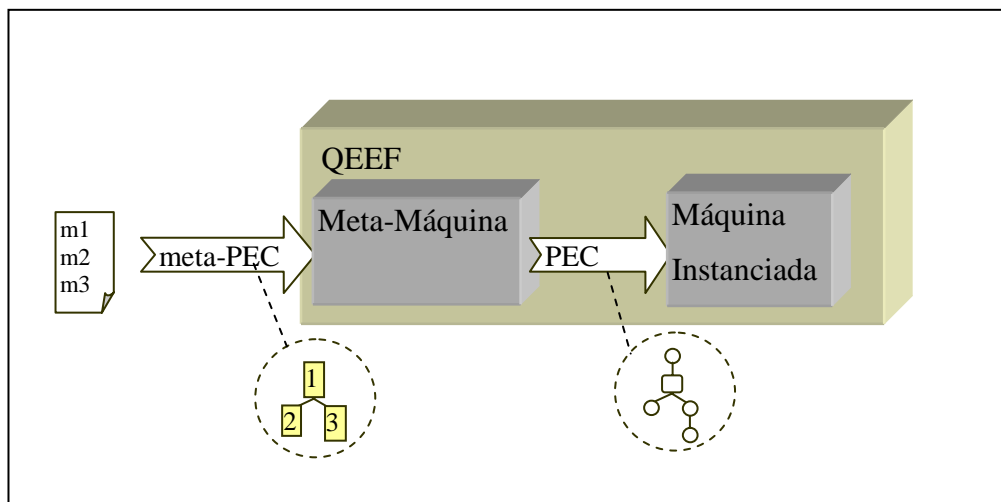


Figura 40 - Os dois níveis de execução do QEEF

5.2. Especificação de meta-PEC em XML

A especificação dos PECs e meta-PECs deve ser elaborada em um documento XML que será validado de acordo com a DTD mostrada no Quadro 7. A principal diferença entre eles está no fato de que o documento XML de um meta-PEC deve ser iniciado pelo elemento <metaplano> (raiz) e o documento XML de um PEC deve ser iniciado pelo elemento <plano>.

Observa-se nesta DTD que o elemento <metaplano> contém o elemento <plano>. Isto se deve ao fato de que um meta-PEC contém um plano algébrico (um PEC) além dos módulos. Para ilustrar a organização desta DTD através de diagramas⁴, a Figura 41 mostra os elementos a partir da raiz do documento, a Figura 42 mostra o elemento <MODULO> e a Figura 43 mostra os elementos <ALGEBRICO> e <ADAPTIVE>.

Para permitir a combinação de módulos e operadores algébricos (elemento <ALGEBRICO>) compondo a árvore do meta-PEC, define-se o elemento <OP> que representa tanto um módulo quanto um operador algébrico.

⁴ Esses diagramas foram projetados utilizando-se a ferramenta de autoria *XMLspy*.

Vê-se que os módulos não são especificados da mesma maneira. Por exemplo, o módulo *Inter-operator* (elemento <INTER>) contém necessariamente dois operadores (que podem ser módulos, operadores algébricos ou um de cada). O módulo *Adaptive*, possui uma lista de operadores (<LISTA>) a serem adaptados, porém, há necessidade de se definir um operador a mais (do qual este módulo consumirá dados).

Exemplos de meta-PECs podem ser encontrados nos estudos de casos do próximo capítulo.

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT METAPLANO (listadeoperadores, MODULO)>
<!ELEMENT MODULO (FIXED | ADAPTIVE | DEMAND-DRIVEN | DATA-DRIVEN |
REMOTE | LOCAL | INTRA | INTER | LASTTUPLE | FIRSTTUPLE | WAITALL
| NOWAIT | WAIT | DEFAULT)>
<!ELEMENT ADAPTIVE ((MODULO | ALGEBRICO), (MODULO | ALGEBRICO)+)>
<!ELEMENT DATA-DRIVEN (MODULO | ALGEBRICO)>
<!ELEMENT DEMAND-DRIVEN (MODULO | ALGEBRICO)>
<!ELEMENT FIRSTTUPLE (MODULO | ALGEBRICO)>
<!ELEMENT FIXED (MODULO | ALGEBRICO)>
<!ELEMENT INTER ((MODULO | ALGEBRICO)+, ALGEBRICO)>
<!ELEMENT INTRA (MODULO | ALGEBRICO)>
<!ELEMENT LASTTUPLE (MODULO | ALGEBRICO)>
<!ELEMENT LOCAL (MODULO | ALGEBRICO)>
<!ELEMENT NOWAIT (MODULO | ALGEBRICO)>
<!ELEMENT REMOTE (MODULO | ALGEBRICO)>
<!ELEMENT WAIT (MODULO | ALGEBRICO)>
<!ELEMENT WAITALL (MODULO | ALGEBRICO)>
<!ELEMENT DEFAULT (ALGEBRICO)>
<!ELEMENT ALGEBRICO (MODULO | ALGEBRICO)*>
<!ATTLIST ALGEBRICO
    classe CDATA #REQUIRED
    ref CDATA #REQUIRED
>
<!ELEMENT listadeoperadores (operador+)>
<!ELEMENT operador (parametro+)>
<!ATTLIST operador
    id CDATA #REQUIRED
    classe CDATA #REQUIRED
>
<!ELEMENT parametro (itemparametro+)>
<!ATTLIST parametro
    tipo CDATA #REQUIRED
>
<!ELEMENT itemparametro EMPTY>
<!ATTLIST itemparametro
    tipo CDATA #REQUIRED
>
```

Quadro 7 - DTD do meta-modelo QUEM (quem.dtd)

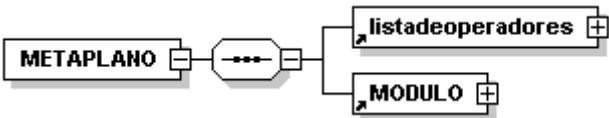


Figura 41 - Elemento metaplano (raiz)

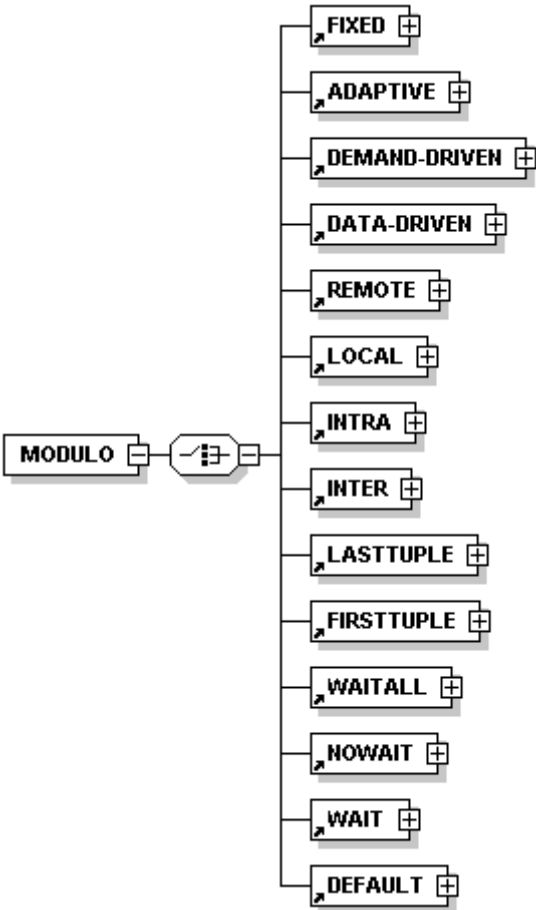


Figura 42 - Detalhamento do elemento MODULO

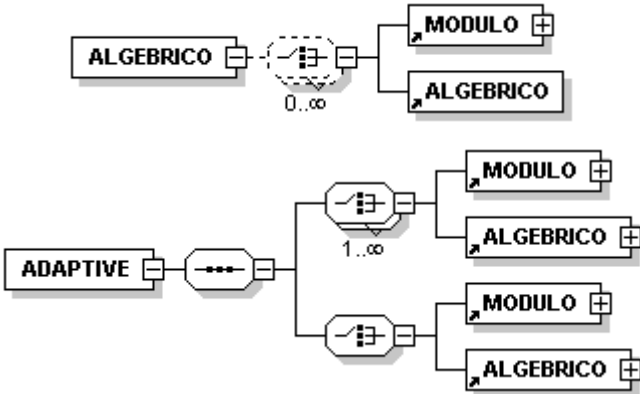


Figura 43 - Detalhamento do modulo Adaptive e do operador Algebrico

5.3. Síntese do Capítulo

Neste capítulo, foi apresentado o meta-modelo QUEM que se baseia na combinação de módulos dentro de um PEC para produzir diferentes modelos de execução. Os PECs especificados segundo o QUEM são denominados de meta-PECs e são pré-processados pelo QEEF (meta-máquina) e, portanto, independem da MEC instanciada, resultando em PECs finais que, por sua vez, são executados pela MEC instanciada gerando o resultado da consulta. Neste contexto, há dois níveis de execução: execução de meta-PECs e de PECs. A especificação dos PECs e meta-PECs deve ser elaborada em um documento XML que será validado de acordo com uma DTD. A principal diferença nesta especificação está no fato de que um meta-PEC deve ser iniciado pelo elemento <metaplano> (raiz) e o um PEC deve ser iniciado pelo elemento <plano>.