4 Uma Linguagem para Workflows de Sistemas de Gerência de Análises em Biossequências

4.1 Introdução

Este capítulo apresenta a linguagem utilizada pelo SGWBio para descrever workflows de Bioinformática.

A linguagem é especificada através de um XML Schema [W3C, 2004b] de tal forma que a descrição de um workflow deverá ser sob forma de um documento XML [W3C, 2004c] satisfazendo o XML Schema. Esta linguagem diz respeito a todos os ambientes de trabalho em que o sistema possa ser implementado.

Optamos por definir uma linguagem de workflow própria por várias razões. Primeiro, a linguagem é bastante simples, não criando um esforço grande cognitivo para ser aprendida.

Segundo, embora fosse possível formular a descrição dos workflows em OWL, como indicado no anexo, preferimos não nos comprometer com nenhuma linguagem para descrever ontologias. De fato, como o SGWBio é um *framework*, a ontologia pode ser definida de diversas maneiras e, portanto, convém adotar uma estratégia neutra com relação à definição dos workflows.

Terceiro, embora já existam diversas linguagens, como BPEL4WS [Weerawarana, 2002], OWL-S [DAML, 2004], XPDL [WfMC, 2002] e XML Pipeline Definition Language [W3C, 2004d], para descrever workflows, em diversos estágios de padronização, após um estudo detalhado das principais, verificamos que nenhuma delas seria inteiramente adequada. Por um lado, as linguagens estudadas ofereciam muito mais recursos de controle do que o necessário aos workflows de Bioinformática. Por outro lado, tais linguagens não cobriam todas as nuances dos dados considerados em Bioinformática, como a leitura e geração de dados gradativa e não gradativa e o tamanho dos contêineres. Portanto, entre adequar alguma linguagem existente e definir uma já sintonizada às nossas necessidades, preferimos a segunda opção.

4.2 Especificação da Linguagem de Workflow em XML Schema

A Figura 6 apresenta diagramaticamente o XML Schema especificando a linguagem, enquanto que o Quadro 7, apresenta o esquema em detalhe. O restante desta seção discute cada elemento da linguagem em detalhes.

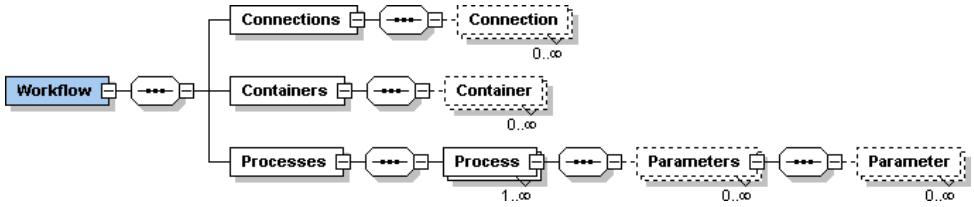

Figura 6. Representação da linguagem do workflow de Bioinformática.

O elemento *Workflow* é a raiz do documento XML do workflow e possui como atributos *name* e *description*, que indicam o nome e a descrição do workflow, definidos pelo pesquisador. Este elemento contém três sub-elementos, *Containers, Processes* e *Connections*.

O elemento *Containers* (Quadro 8) é composto de vários sub-elementos *Container*, representante do contêiner utilizado para a conexão entre as instâncias de processos.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--W3C Schema generated by XML Spy v4.3 U (http://www.xmlspy.com)-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Workflow">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Connections">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Connection" minOccurs="0" maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:attribute name="name" type="xs:string" use="required"/>
                                    <xs:attribute name="type" use="required">
                                        <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                                <xs:enumeration value="gradative"/>
                                                <xs:enumeration value="not_gradative"/>
                                            </xs:restriction>
                                        </xs:simpleType>
                                    </xs:attribute>
                                    <xs:attribute name="source" type="xs:string" use="required"/>
                                    <xs:attribute name="target" type="xs:string" use="required"/>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Containers">
                    <xs:complexType>
                        <xs:sequence>
```

```
                <xs:element name="Container" minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:attribute name="name" type="xs:string" use="required"/>
                        <xs:attribute name="type" use="required">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:enumeration value="gradative"/>
                                    <xs:enumeration value="not_gradative"/>
                                    <xs:enumeration value="mix"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="data_class" type="xs:string" use="required"/>
                        <xs:attribute name="data_format" type="xs:string" use="required"/>
                        <xs:attribute name="min_size" type="xs:string" use="optional"/>
                        <xs:attribute name="max_size" type="xs:string" use="optional"/>
                        <xs:attribute name="file_path" type="xs:string" use="optional"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Processes">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Process" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Parameters"
                                        minOccurs="0"
                                        maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="Parameter"
                                                    minOccurs="0"
                                                    maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:attribute name="name"
                                                              type="xs:string"
                                                              use="required"/>
                                                <xs:attribute name="type"
                                                              type="xs:string"
                                                              use="required"/>
                                                <xs:attribute name="default_value"
                                                              type="xs:string"
                                                              use="optional"/>
                                                <xs:attribute name="config_value"
                                                              type="xs:string"
                                                              use="required"/>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                        <xs:attribute name="id" type="xs:ID" use="required"/>
                        <xs:attribute name="name" type="xs:string" use="required"/>
                        <xs:attribute name="type" use="required">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:enumeration value="constructive"/>
                                    <xs:enumeration value="internal_control"/>
                                    <xs:enumeration value="external_control"/>
                                    <xs:enumeration value="filter"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:attribute>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
```

```
            </xs:element>
        </xs:sequence>
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="description" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
</xs:schema>
```

Quadro 7. XML Schema do Workflow.

O elemento *Container* possui os atributos *name* e *type*, que indicam, respectivamente, o nome que identifica o contêiner e o seu tipo, considerando que ele deve estar adequado para a conexão de instâncias de processos que podem ler e gerar dados de formas diferentes: não-gradativa ou gradativa, assunto tratado no Capítulo 3 e que será detalhado a frente. Além destes atributos, o *Container* tem ainda os atributos: *data_class* e *data_format*, que indicam, respectivamente, a classe e o formato do dados que armazena, *min_size* e *max_size*, que indicam os tamanhos mínimo e máximo estimados para armazenar os dados, e *file_path*, que é a localização de dados que podem já estar armazenados em disco, antes mesmo da execução do workflow.

```
<xs:element name="Containers">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Container" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="name" type="xs:string" use="required"/>
                    <xs:attribute name="type" use="required">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="gradative"/>
                                <xs:enumeration value="not_gradative"/>
                                <xs:enumeration value="mix"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                    <xs:attribute name="data_class" type="xs:string" use="required"/>
                    <xs:attribute name="data_format" type="xs:string" use="required"/>
                    <xs:attribute name="min_size" type="xs:string" use="optional"/>
                    <xs:attribute name="max_size" type="xs:string" use="optional"/>
                    <xs:attribute name="file_path" type="xs:string" use="optional"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Quadro 8. Workflow em XML-Schema: Elemento *Containers*.

O elemento *Processes* é composto de sub-elementos *Process*, que representam as instâncias de processos do workflow. O elemento *Process* possui os atributos *id*, *name* e *type*, sendo que *id* é a identificação da instância de processo, *name* é o nome da instância do processo e *type* é o tipo de processo, podendo ter os seguintes valores:

- *constructive* para os processos construtivos;
- *filter* para para os processos filtros;
- *internal_control*, para os processos de controle interno do workflow.
- *external_control*, para os processos de controle externo de execução do workflow.

Além disto, o elemento *Process* (Quadro 9) também possui o sub-elemento *Parameters,* que representam os parâmetros do processo.

```
<xs:element name="Processes">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Process" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Parameters" minOccurs="0" maxOccurs="unbounded">
                        <!-- Definidos a seguir -->
                        </xs:element>
                    </xs:sequence>
                    <xs:attribute name="id" type="xs:ID" use="required"/>
                    <xs:attribute name="name" type="xs:string" use="required"/>
                    <xs:attribute name="type" use="required">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="constructive"/>
                                <xs:enumeration value="internal_control"/>
                                <xs:enumeration value="external_control"/>
                                <xs:enumeration value="filter"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Quadro 9. Workflow em XML-Schema: Elemento *Processes*.

O elemento *Parameters* (Quadro 10) é composto de sub-elementos *Parameter*, que representam cada parâmetro da instância de processo e possui os seguintes atributos:

- *name*: a identificação do parâmetro;

- *type*: a identificação do tipo de dados do parâmetro como, por exemplo, inteiro ou cadeia de caracteres;
- *default_value*: o valor padrão para o parâmetro, ou seja, o valor já configurado pela instância de processo;
- *config_value*: o valor para o parâmetro definido pelo pesquisador; importante no caso do pesquisador desejar executar a instância de processo com valores diferentes dos valores *default_value*.

```
<xs:element name="Parameters" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Parameter" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="name" type="xs:string" use="required"/>
                    <xs:attribute name="type" type="xs:string" use="required"/>
                    <xs:attribute name="default_value" type="xs:string" use="optional"/>
                    <xs:attribute name="config_value" type="xs:string" use="required"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Quadro 10. Workflow em XML-Schema: Elemento *Parameters*.

O elemento *Connections* é composto de sub-elementos *Connection* (Quadro 11), que representa uma conexão entre um processo e um contêiner, e possui os seguintes atributos:

- *name*: identificador da conexão;
- *type*: tipo da conexão, podendo ter os valores *gradative* ou *not-gradative*;
- *source*: que assume valores dos identificadores de um elemento *Container* ou *Process*;
- *target*: que assume valores de um elemento *Process*, se o valor de *source* for um elemento *Container*, ou de um elemento *Container*, se o valor de *source* for um elemento *Process*.

```
<xs:element name="Connections">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Connection" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="name" type="xs:string" use="required"/>
                    <xs:attribute name="type" use="required">
                        <xs:simpleType>
```

```
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="gradative"/>
                            <xs:enumeration value="not_gradative"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="source" type="xs:string" use="required"/>
                <xs:attribute name="target" type="xs:string" use="required"/>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
```

Quadro 11. Workflow em XML-Schema: Elemento *Connections*.

Analisando este documento, nota-se que é possível a construção do grafo que representa o workflow através da criação de nós para os processos, nós para os contêineres e de arcos que conectam os processos e contêineres.

Como exemplo, segue um grafo (Figura 7) e um documento em XML (Quadro 12) definindo um workflow simples, que ilustra a composição dos programas Phred e Phrap, parte inicial do workflow da Figura 1.

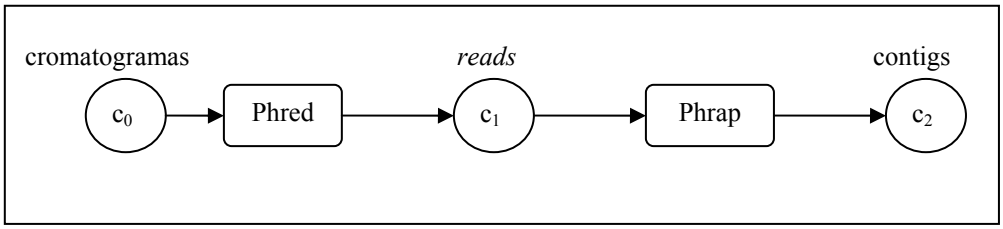

Figura 7. Exemplo de grafo bipartido de workflow.

```
<?xml version="1.0" encoding="UTF-8"?>
<Workflow>
 <Containers>
  <Container
        name="Container_0"
        type ="gradative"
        data_class =" chromatogram_set"
        data_format =" chromatogram_set_abi_format "
        file_path =" c:/chromat_dir"/>
  <Container
        name =" Container_1"
        type ="mix"
        data_class ="read_nucleotide_sequence_set"
        data_format ="sequence_set_fasta_format" />
  <Container
        name =" Container_2"
        type ="gradative"
        data_class ="contig_nucleotide_sequence_set "
        data_format ="sequence_set_fasta_format " />
 </Containers>

<Processes>
<Process name ="phred" id="Process_1"  type ="constructive">
```

```
    <Parameters>
      <Parameter name ="nocall"  default_value="false" config_value ="false" />
      <Parameter name ="trim"  default_value ="false" config_value ="true" />
    </Parameters>
</Process>

<Process name="phrap"  id="Process_2"  type =" constructive">
    <Parameters>
      <Parameter name ="forcelevel"  default_value ="0" config_value ="1" />
      <Parameter name ="maxgap"  default_value ="30" config_value ="20" />
    </Parameters>
</Process>
 </Processes>

<Connections>
<Connection
        name ="Connection_1"
        type ="gradative"
        source ="Container_0"
        target =" Process_1"/>
<Connection
        name ="Connection_2 "
        type ="gradative"
        source ="Process_1"
        target ="Container_1"/>
<Connection
        name ="Connection_3"
        type ="not_gradative"
        source ="Container_1"
        target ="Process_2"/>
<Connection
        name ="Connection_4"
        type ="gradative"
        source ="Process_2"
        target ="Container_2 "/>
</Connections >

 </Workflow>
```

Quadro 12. Exemplo de documento de especificação de workflow.

4.3 Comentários Finais

Este capítulo apresentou a especificação (em XML Schema) da linguagem de workflow utilizada pelo SGWBio para definir documentos (em XML) de especificação de workflows em Bioinformática.

O documento de especificação refletirá o workflow definido por um pesquisador e conterá informações adicionais, como instâncias de processos extras, que permitirão que o workflow seja executado de forma coerente e otimizada.

A utilização do documento de especificação do workflow pelo SGWBio será apresentada em detalhes nos próximos capítulos.