6
Referências Bibliográficas

[1] LINTHICUM, D. **Enterprise application integration**. [S.l.]: Addison-Wesley Longman Ltd. Essex, UK, UK, 2000. 1

[2] ERL, T. **Service-oriented architecture: concepts, technology, and design**. [S.l.]: Prentice Hall PTR Upper Saddle River, NJ, USA, 2005. 1

[3] HOHPE, G.; WOOLF, B. **Enterprise integration patterns: Designing, building, and deploying messaging solutions**. [S.l.]: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2003. 1

[4] MOORE, R. Data management systems for scientific applications. **Proceedings of The Architecture of Scientific Software**, p. 273–284, 2000. 1

[5] GRAY, J. et al. Scientific data management in the coming decade. **ACM SIGMOD Record**, ACM New York, NY, USA, v. 34, n. 4, p. 34–41, 2005. 1

[6] LUDÄSCHER, B. et al. C.(2006). Managing scientific data: From data integration to scientific workflows. **Geoinformatics: Data to Knowledge, GSA Special Paper**, v. 397, p. 109–130. 1

[7] GROUP, T. T. H. **HDF Group - HDF5**. Disponível em: <http://hdf.ncsa.uiuc.edu/HDF5>. 1, 2

[8] BOLTON, F.; WALSHE, E. **Pure CORBA: A Code-Intensive Premium Reference**. [S.l.]: Sams Indianapolis, IN, USA, 2001. 1, 2.5

[9] UNIDATA. **NetCDF (network Common Data Form)**. Disponível em: <http://www.unidata.ucar.edu/software/netcdf>. 2

[10] Google Inc. **Protocol Buffers - Google's data interchange format**. Disponível em: <http://code.google.com/p/protobuf>. 2, 2.4

[11] The OpenSpirit Corporation. **OpenSpirit - Products**. 2006. Disponível em: <http://www.openspirit.com/products.html>. 2, 2.3

[12] MCGRATH, R. **E. A Scientific Data Server: The Conceptual Design. White Paper**. [S.l.]: NCSA, University of Illinois, Urbana-Champaign, 1997. 2, 2.1

[13] Open SOA Collaboration. **Service Data Objects Specifications**. 2007. Disponível em: <http://www.osoa.org/display/Main/Service+Data+Objects+Specifications>. 2.2

[14] SZYPERSKI, C. Component technology: what, where, and how? In: IEEE COMPUTER SOCIETY WASHINGTON, USA. **Proceedings of the 25th International Conference on Software Engineering**. [S.l.], 2003. p. 684–693. 3

[15] AUGUSTO, C. et al. **SCS: Software Component System**. maio 2007. Disponível em: <http://www.tecgraf.puc-rio.br/ scorrea/scs>. 3

[16] LIMA, M. J. et al. Csbase: A framework for building customized grid environments. In: **Proceedings of WETICE '06**. Washington, USA: IEEE Computer Society, 2006. p. 187–194. ISBN 0-7695-2623-3. 4.1.2

[17] BAKER, J. H. C.; HEWITT, C. The incremental garbage collection of processes. In: LOW, J. (Ed.). **Proceedings of the Symposium on Artificial Intelligence and Programming Languages**. New York, USA: ACM Press, 1977. p. 55–59. 4.2

A
Apêndice

A.1
IDL do Serviço de Dados

A listagem A.1 apresenta a IDL da arquitetura de servidores de dados. Nesta listagem estão definidos todos os tipos utilizados na arquitetura, incluindo o *IDataService*, que é o tipo do serviço de dados. Alguns tipos de dados básicos, que devem ser utilizados por várias visões de diferentes domínios, são definidos na listagem A.2

A.2
IDL do SCS

A listagem A.3 apresenta uma versão resumida da IDL do SCS. Nessa listagem são apresentados apenas os tipos usados na arquitetura de servidores de dados.

A.3
IDL dos Dados de Poço

Os dados do estudo de caso de poços de petróleo estão definidos na listagem A.4. As duas visões utilizadas, *corporativa* e *científica* são apresentadas, bem como uma visão abstrata que contém os dados comuns às duas visões utilizadas. A visão corporativa foi utilizada também nos experimentos utilizados para avaliar o desempenho da arquitetura.

A.4
Arquivos proto com as definições de dados de poço

Os experimentos com o *Protocol Buffers* utilizaram descrições dos tipos de dado escritas na IDL específica dessa tecnologia. Foi necessário, portanto, redefinir o tipo de poço científico e todos os tipos relacionados que fazem parte da arquitetura para uma representação própria do *Protocol Buffers*.

Os tipos redefinidos do SCS, estão na listagem A.5. Os tipos que fazem parte da definição da arquitetura estão nas listagens A.6 e A.7. Por fim, os tipos ligados ao dado de poço, estão na listagem A.8.

Listagem A.1: IDL do Serviço de Dados

```
#ifndef __SDEAC_IDL__
#define __SDEAC_IDL__

#include <scs.idl>

module sdeacidl {

module types {
  typedef sequence<string> StringSeq;
};

typedef string URI;
struct DataKey {
  scs::core::ComponentId fServerID;
  URI fActualDataID;
};
typedef sequence<DataKey> DataKeyList;

struct Metadata {
  string fName;
  string fValue;
};
typedef sequence<Metadata> MetadataList;

valuetype DataDescription {
  public DataKey fKey;
  public sdeacidl::types::StringSeq fViews;
  public MetadataList fMetadata;
};
typedef sequence<DataDescription> DataDescriptionList;

module service {
  interface IDataService;
};

abstract interface DataView {
  DataKey getKey();
  service::IDataService getDataService();
  string getViewInterface();
};
typedef sequence<DataView> DataViewList;

module service {

exception OperationNotSupported {};
exception UnknownType {};

interface IDataService {
  DataDescriptionList getRoots();
  DataDescriptionList getChildren(in DataKey fKey);
  DataKey createData(in DataKey fParentKey, in MetadataList fMetadata)
    raises (OperationNotSupported);
  DataKey createDataFrom(in DataKey fParentKey, in DataKey fSourceKey)
    raises (OperationNotSupported, UnknownType);
  boolean deleteData(in DataKey fKey) raises (OperationNotSupported);
  DataDescription getDataDescription(in DataKey fKey);
  DataView getDataView(in DataKey fKey, in string fViewInterface);
  DataViewList getDataViewList(in DataKeyList fKeyList,
      in string fViewInterface);
  sdeacidl::types::StringSeq getViewInterfaces(in DataKey fVey);
};

};

};

#endif
```

Listagem A.2: IDL do Serviço de Dados - Tipos Básicos

```
#ifndef __SDEAC_TYPES_IDL__
#define __SDEAC_TYPES_IDL__

#include <sdeac.idl>

module sdeacidl {

module types {

typedef double Number;
typedef sequence<Number> NumberList;

struct Point {
  float fX;
  float fY;
};
typedef sequence<Point> PointList;

valuetype UnstructuredData supports sdeacidl::DataView {
  public string fHost;
  public unsigned long fPort;
  public boolean fWritable;
};

};

};
```

Listagem A.3: IDL do SCS

```
#ifndef SCS_IDL
#define SCS_IDL

module scs {

module core {

exception StartupFailed {};
exception ShutdownFailed {};

struct ComponentId {
  string name;
  octet major_version;
  octet minor_version;
  octet patch_version;
};
typedef sequence<ComponentId> ComponentIdSeq;

interface IComponent {
  void startup() raises (StartupFailed);
  void shutdown() raises (ShutdownFailed);

  ComponentId getComponentId ();

  Object getFacet (in string facet_interface);
  Object getFacetByName (in string facet);
};

};

};

#endif
```

Listagem A.4: IDL dos Dados de Poço

```
#ifndef __WELL_IDL__
#define __WELL_IDL__

#include <sdeac.idl>
#include <sdeac/types.idl>

module wellidl {

typedef sdeacidl::MetadataList Header;

abstract valuetype Well {
  Header getHeader();
  sdeacidl::types::Point getLocation();
};

enum Situation {
  LEASE, DRILLING, DRILLED
};

enum Classification {
  SOME
};

valuetype EnterpriseWell : Well supports sdeacidl::DataView {
  public string fName;
  public Situation fSituation;
  public unsigned long fANPCode;
  public Classification fClassification;
};

struct Curve {
  string fName;
  Header fHeader;
  sdeacidl::types::NumberList fValues;
};
typedef sequence<Curve> CurveList;

valuetype ScientificWell : Well supports sdeacidl::DataView {
  public CurveList fCurves;
};

};

#endif
```

Listagem A.5: Arquivo proto do SCS

```
option optimize_for = SPEED;

package scsproto;

message ComponentId {
  required string name = 1;
  required uint32 major_version = 2;
  required uint32 minor_version = 3;
  required uint32 patch_version = 4;
}
```

Listagem A.6: Arquivo proto do Serviço de Dados

```
option optimize_for = SPEED;

package sdeacproto;

import "scs_def.proto";

message DataKey {
  required scsproto.ComponentId server_id = 1;
  required string actual_data_id = 2;
}

message DataView {
  required DataKey key = 1;
  required string data_service_ior = 2;
  required string view_interface = 3;
}
```

Listagem A.7: Arquivo proto do Serviço de Dados - Tipos Básicos

```
option optimize_for = SPEED;

package sdeacproto.types;

message Point {
  required float x = 1;
  required float y = 2;
}

message PointList {
  repeated Point value  = 1;
}
```

Listagem A.8: Arquivo proto dos Dados de Poço

```
option optimize_for = SPEED;

package wellproto;

import "sdeac_def.proto";
import "sdeac/types_def.proto";

message Well {
  required sdeacproto.DataView dataView = 1;
  required sdeacproto.MetadataList header = 2;
  required sdeacproto.types.PointList pointList = 3;
};

message Curve {
  required sdeacproto.MetadataList header = 1;
  repeated double values = 2;
}

message CurveList {
  repeated Curve curve = 1;
}

message ScientificWell {
  required Well well = 1;
  required CurveList curveList = 2;
}
```