Finding My CRS: A Systematic Way of Identifying CRSs

Peter Baumann
Jacobs University Bremen
28759 Campus Ring 1
Bremen, Germany
p.baumann@jacobs-university.de

Piero Campalani
Università degli Studi di Ferrara
Via Saragat 1
44122 Ferrara, Italy
cmppri@unife.it

Dimitar Misev
Jacobs University Bremen
28759 Campus Ring 1
Bremen, Germany
d.misev@jacobs-university.de

Jinsongdi Yu
Jacobs University Bremen
28759 Campus Ring 1
Bremen, Germany
j.yu@jacobs-university.de

ABSTRACT

In the era of Big Data, being able to work with multidimensional arrays in a robust and consistent manner as the number and variety of dimensions increase, is just as important as being able to handle the large volumes inherent to this type of data. Usually, array analytics is carried out to extract meaningful information for further applications, e.g. slicing and subsetting. While domain-specific dimensions, which are beyond spatio-temporal, underlie rich domain anchor semantics, assigning consistent dimension schema for Points Of Interest (POI) across multidisciplinary data sets is challenging. New compositions of CRSs need to be constructed on the fly by a heterogeneous community with different backgrounds and applications in mind, consequently, linking dimensions via different resolvers to drive away dimension fragments from high-dimensional spaces. We propose to identify dimensions via a linked resolver approach. Such an approach allows CRSs to be referred to and looked up across multidisciplinary applications. Finally, we present a planetary use case, and specification- and scenario-based testing results to validate our approach.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services*; J.2 [**Computer Applications**]: Physical Sciences and Engineering—*Astronomy, Earth and atmospheric sciences*

General Terms

Theory, Standardization

Keywords

Coordinate Reference System, URI, multi-dimensional data, OGC

1. INTRODUCTION

Coordinate Reference Systems (CRSs) are at the heart of all geo data, as they determine the meaning of coordinate positions relating object locations in space and time to some reference position. Every spatially referenced object needs to carry along, together with its location, the CRS in which this location is expressed.

The wider availability of free spatial data from satellite sensors (LANDSAT, MODIS, SRTM, ASTER, etc.), the quick development of more and more powerful FOSS GIS systems and software packages for spatial analysis (e.g. [7, 8, 22]), and the well-known continuous growth of disk space and processing power of common laptop and desktop computers are causing a rapid expansion of the scientific geospatial community, worldwide. This surely brings huge benefits in the development of any related research field, from simple monitoring to modelling and forecasting applications. On the other hand, the immediate availability of these high-level tools is letting in people with insufficient expertise on the basics of datums and projections, partially – sometimes totally – missing the required learning curve. Much more attention to the meaning of CRSs must be then addressed, and the proposed schema can provide a solution to gradually put the community closer into contact with the *content* of CRSs.

As typically some standardized, well-known CRS is used, the CRS definition itself does not have to be stored and transmitted; rather, it is sufficient to remember the identifier. The EPSG database [3] defines such a set of CRSs relying on a numerical system – for example, WGS84 has the EPSG code 4326. Any necessary projection into another CRS classically is hardwired into the GIS code.

For a long time, when horizontal maps were served only, this was enough. With the advent of Web services offering new information categories, however, this has changed. 3-D and 4-D data sets from atmospheric and ocean sciences, geology, astronomy, and many more domains add further CRS

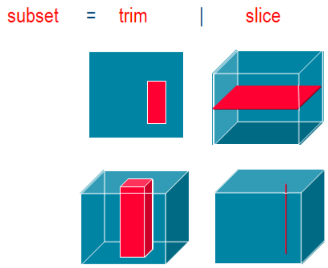

Figure 1: WCS subsetting types: trimming and slicing

axes beyond latitude and longitude. Vertical axes like depth and elevation require new CRSs; while there is a WGS-3D in EPSG, by far not all combinatorially possible combinations of horizontal and vertical axes is supported. Time gets added as an axis, too. Traditionally, this is handled very differently, contributing significantly to today's difficulties in large-scale time series analysis. There is no EPSG mentioning of time, rather ISO 8601 [14] comes into play here. Additionally, the "Auto CRS" concept introduced by OGC Web Map Service [9] is standalone and not integrated with general CRS handling.

A good example for the problems arising from this new wealth of dimensionality is constituted by raster data, more generally: coverages as standardized in [12] and [5]. Modern Web service standards like the Web Coverage Service (WCS) [20] and Web Coverage Processing Service (WCPS) [19, 6] offer versatile slicing methods where the dimension of a data cube is reduced (Figure 1). This leads to the situation that often there is no predefined CRS matching the result coverage – for example, x/t and y/z combinations are not covered by EPSG. Hence, it is not clear what CRS identifier such a result coverage should contain when sent back to the client.

For all such CRSs, a simple, expressive, and Web compatible mechanism is required allowing software to create, identify, and understand CRSs and their definitions. Within OGC, this has initiated discussion on a unified handling of CRSs. Definition of CRSs is given by the Geography Markup Language (GML) [13], but a unified referencing of the various CRSs is not present yet. The result is the OGC CRS Name Type Specification (NTS) which has been discussed among many stakeholder communities within and beyond OGC and, following general acceptance, is about to be published by OGC. It relies on URLs as identification mechanisms which extends the EPSG URLs in a natural way.

In this contribution, we present core concepts of this CRS NTS. Section 2 introduces the concept of CRS identifiers; practical use cases along with the benefits of this new CRS handling are presented in Section 3, whereas in Section 4 the testing issue is treated. Conclusions are finally drawn in Section 5.

2. CRS IDENTIFIERS

Coordinate Reference Systems (CRSs) are used to unambiguously describe the location of objects in space and time. The CRS itself is defined by a coordinate system which sets properties like axis sequence and units of measure, and a datum which fixes locations to a reference object.

CRSs can be described in detail using languages like the Well-Known Text (WKT), or the Geography Markup Language (GML). Communicating complete definitions in order to identify a CRS is rather cumbersome though, and a unique identifier is preferred – even more as the vast majority of applications rely on some commonly accepted standard CRSs.

A Spatial Reference System Identifier (SRID) is a unique value used to unambiguously identify CRSs. Virtually all major spatial vendors have created their own SRID implementation or refer to those maintained by an authority, such as the Oil and Gas Producers (OGP) Surveying and Positioning Committee (formerly maintained by the European Petroleum Survey Group). The EPSG SRID value for the WGS84 CRS is 4326 for example. In OGC, SRID codes have been initially used in URNs conforming to a general scheme of identifying concepts within OGC [OGC 07-092r1]:

```
"urn:ogc:def" ":" objectType ":" authority ":" [ version ] ":" code
```

For example, the URN identifying WGS84 could look like:

```
urn:ogc:def:crs:EPSG::4326
```

Identifying CRSs via such URNs is not suitable in all cases. Especially when it comes to non-standard CRSs for multidimensional data, a more flexible and scalable approach is necessary. Therefore, we propose a Name Type Specification (NTS), which allows to identify predefined, combined, concatenated and parametrized CRS definitions via URLs. A registry service able to resolve such URLs to CRS definitions represented in GML (and further planned WKT) has already been implemented.

2.1 Predefined definitions

Identifiers for predefined CRSs within a certain authority are the URL alternative of the deprecated OGC URNs. Such an identifier consists of three components (cf. Table 1): the authority that maintains the CRS, the version, and the CRS code. Consider for example the identifier for the WGS84 CRS:

```
http://www.opengis.net/def/crs/EPSG/0/4326
```

This URL directly resolves to the GML definition of the WGS84 CRS. Besides the RESTful URL style shown above, the same identifier can be represented in a key-value (KVP) query format, allowing for more flexibility:

```
http://www.opengis.net/def/crs?authority=EPSG
&version=0&code=4326
```

Besides CRS definitions, further entities like datums, meridians, coordinate systems, etc. are also referenceable. An example identifier for the Greenwich prime meridian:

```
http://www.opengis.net/def/meridian/EPSG/0/8901
```

2.2 Parameterized CRSs

The WMS specification [9] defines a special class of "automatic" coordinate reference systems that include a user-selected centre of projection. Here we extend this concept to

Table 1: Identifier parameters

Name	Definition	Data type
authority	the OGC-specified abbreviation for the authority organization that specified the referenced definition. As such, it identifies an authority recognized by the OGC, for example "EPSG" or "ISO"	NCName
version	the version of the authority or code for the referenced definition. When the referenced definition does not have a version use the string "0" (without quotes).	String
code	unique identifier of the CRS definition itself, as specified by the referenced authority, for example "4326" is the EPSG code for WGS84.	NCName

generic *parameterized* CRSs, where it is possible to instantiate an abstract, or template CRS definition to a concrete and unique CRS based on user-supplied parameter values, and optionally parameters derived from these via some mathematical formula. The example below shows a parameterized CRS that defines the auto universal transverse mercator layer CRS (AUTO2:42001) as specified in [9].

```
<ParameterizedCRS
  xmlns="http://www.opengis.net/crs-nts/1.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  gml:id="parameterized-crs-4326">
  <gml:identifier>
    http://www.opengis.net/def/crs/AUTO/1.3/42001
  </gml:identifier>
  <gml:scope>not known</gml:scope>
  <parameters>
    <parameter name="lon"/>
    <parameter name="lat">
      <value>0.0</value>
    </parameter>
    <parameter name="zone">
      <value>
    min( floor( (${lon} + 180.0) / 6.0 ) + 1, 60 )
      </value>
    </parameter>
    <parameter name="central_meridian">
      <value>-183.0 + ${zone} * 6.0</value>
      <target>//greenwichLongitude</target>
    </parameter>
    <parameter name="false_northing">
      <value>
        (${lat} >= 0.0) ? 0.0 : 10000000.0
      </value>
      <target>//falseNorthing</target>
    </parameter>
  </parameters>
  <targetReferenceSystem xlink:href=
    "http://www.opengis.net/def/crs/EPSG/0/4326"/>
</ParameterizedCRS>
```

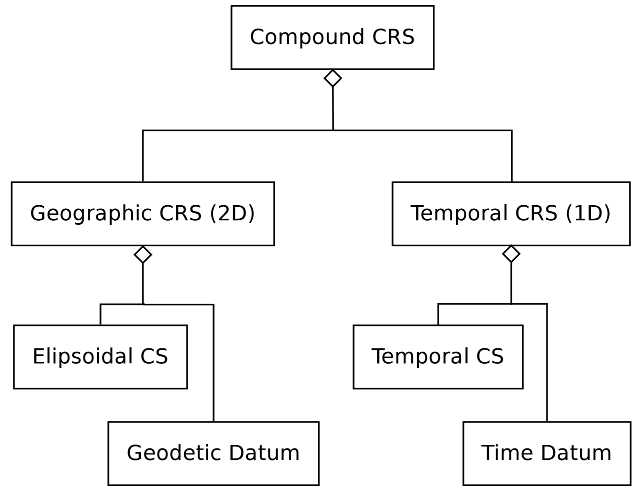

Figure 2: Composition of spatial and temporal coordinate reference systems

In the definition of the parameterized CRS, the concrete CRS is referenced via the `targetReferenceSystem` element. The parameter list defines substitutions that will be done over the target CRS at instantiation time (when the CRS identifier is resolved). Parameter values that will substitute the target elements must be valid JavaScript expressions [10]. By enclosing parameter names in ${ and }, their values can be referenced and used by other parameters (but recursion is not allowed). The value expressions are analyzed and evaluated in such an order that ensures all referenced parameters in an expression have already been evaluated, before submitting it to the JavaScript engine.

2.3 Composing CRSs

Traditionally coordinate reference systems are usually horizontal (2D) or vertical (1D). 3D or higher dimensional points are described by combining horizontal coordinates from one coordinate reference system with height or depth coordinates from another for example. Such a point composed of coordinates from different coordinate reference systems is referenced to a new *compound* CRS, which is composed of the respective, non-repeating, single component CRSs [11]. The coordinate order in the compound CRS follows the order of the coordinates in the component coordinate reference systems.

A URL that identifies such a compound CRS contains identifiers to predefined CRSs (cf. Section 2.1). Since the order of component CRSs matters, it has to be preserved in the compound identifier too, which is done by simply numbering the keys starting from one. So the compound CRS for 3D time-series data could be composed from the 2D WGS84 and ISO 8601 time CRSs:

```
http://www.opengis.net/def/crs-compound?
  1=http://www.opengis.net/def/crs/EPSG/0/4326&
  2=http://www.opengis.net/def/crs/ISO/2004/8601
```

The spatio-temporal compound CRS identified by this URL is schematically presented in Figure 2.

2.4 Derived CRSs

The possibility to parse XML pointed by an URI can become crucial for instance when a dataset relies on its specific *derived* CRS. This is the typical case of ground-based radar imagery: the CRS of such data is rooted in a georeferenced CRS by the location of the radar itself, but a further layer of operations is needed to fully reference the single pixels of each image.

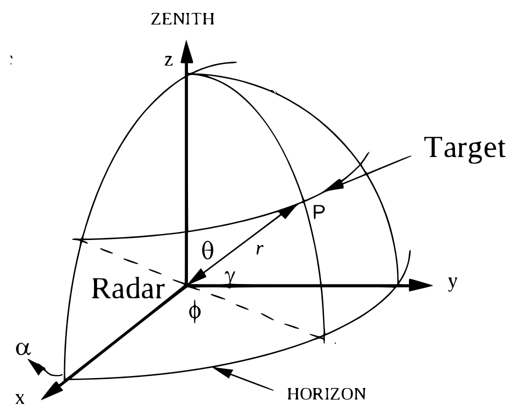

Figure 3: The internal coordinate reference system of a radar.

From an EPSG point of view, there is the need to define a "coordinate operation", which would be a coordinate "conversion" in this case, by specifying the proper "parameter values" for the selected "operation method", which describes a formula along with the required parameters. In this example the conversion is meant to be from the *base* geographic CRS, say WGS84, to the *derived* CRS of the radar, say an internal engineering spherical CRS [3, 4]. The user could then upload its own forward and backward conversion methods for his specific datasets, and then tie them directly to the CRS of the data:

```
http://www.opengis.net/def/crs-concatenated?
  base=http://www.opengis.net/def/crs/EPSG/0/4326&
  forwardOp=http://www.opengis.net/def/
    coordinateOperation/EPSG/0/<code_forw>&
  backwardOp=http://www.opengis.net/def/
    coordinateOperation/EPSG/0/<code_back>
```

This way any server would be able to read the required information to decode this derived CRS, so as for instance to convert from and to radar coordinates: the cartographic library would know the conversion from a specified geodetic/ projected CRS to the base CRS, WGS84 in this case, while the second step of transformation to the spherical CRS of the radar would be achieved by supplying the library with the coordinate conversion parameters and formulas.

3. PRACTICAL EXAMPLES

In this section we are going to show some direct advantages that could ensue from this flexible URI-oriented CRS conception, both for the increased robustness and simplicity in the metadata management issue, and for the extended possibilities that could be exploited in such contexts.

3.1 Beyond spatio-temporal datasets

One of the major advantages that our CRS schema would bring is the general validity that is applied to a reference system, regardless of which kind of axes are composing it, thus going beyond the traditional pure spatial or, at the most, spatio-temporal domains. Building up N-dimensional Compound Coordinate Reference Systems (CCRSs) gives room to high flexibility when there is the need to handle and serve hypercubes of data: a unique schema is applied irrespectively of the dimensionality and semantic of the underlying datasets.

From the perspective of a Web Service, a URI-oriented metadata management additionally means more simplicity and robustness. All the auxiliary information that is indeed required for a proper management of data, being it 1- or multi-dimensional, is transferred to the GML definitions which are behind the CRS URI assigned to it. In the case where the data offered must be untied from any specific context, the metadata database this way loses the challenging responsibility of a consistent management of the data served. The CRS URI would instead hold all that is needed for a service to correctly administer even a highly dimensional dataset: the database gets thiner and more robust at the same time, there is no duplication of metadata, which is instead concentrated in the GML definitions, hence avoiding human-related leakages in the database structure and content. This way the referencing is moved upwards as a *one* single field assigned to the dataset.

As a practical example, a scientific research group might need to serve daily composites of atmospheric profiles which are delivered by the Moderate Resolution Imaging Spectroradiometer (MODIS) sensors aboard the polar-orbiting Terra (EOS AM) and Aqua (EOS PM) satellites[1] over a certain geographic area of interest: to reduce the complexity of this example, we can focus on the publication of just one of the several atmospheric products offered by MODIS, *temperature* for instance[2]. For example, these high-resolution profiles might be exploited for corrections on atmospheric effects for some of the MODIS products themselves, or for clear-sky atmospheric characterization in global greenhouse studies [24, 15, 16, 25]. Specifically, the temperature profiles are delivered as grids at 5×5 km of spatial resolution (at nadir) and at 20 irregularly spaced vertical levels, based on different pressure values, namely from 5 to 1000 hPa. Building up a time series of such profiles would thus mean creating a 4-dimensional collection, having two horizontal axes (easting and northing of each single projected image), the vertical dimension represented by the pressure levels,

[1] With their wide swath (2230 km) and large spectral range (36 channels between 0.412 to 14.2 µm) MODIS products have been heavily used in the research in the last years [23].
[2] In the end, one could either spread the different products onto a single collection of data by means of the *range* of a dataset, or simply build a separate collection for each product.

and finally time.

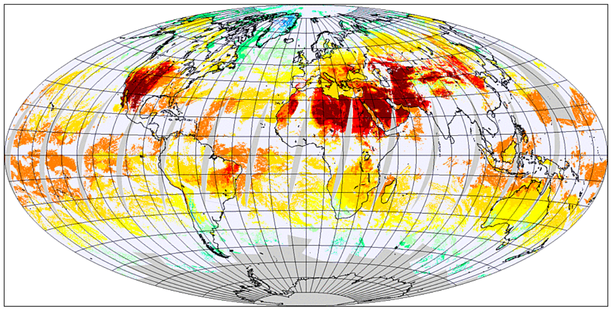

Figure 4: Example of MODIS temperature profile and worldwide daily coverage.

Having stored this huge dataset in a database, then the handling of the metadata need to be faced. The job of a metadata database is to provide a middle layer between the semantic-aware user requests and the typically pure cell-based underlying database indexing. Having the possibility to gather all the auxiliary information into the CRS definition, in our case we would need to store only the dataset-specific information, that means the extents (maximum and minimum value) for each of our four domains and the mapping of each irregularly spaced pressure level to its corresponding cell index. The order of the axis in the database is assumed to correspond to the order intrinsically set by the CCRS of our dataset, which would be for example defined as follows:

```
http://www.opengis.net/def/crs-compound?
  1=http://www.opengis.net/def/crs/EPSG/0/32632&
  2=http://www.opengis.net/def/crs?
    authority=OGC&
    version=0.1&
    code=Linear1D&
    axisName=Pressure&
    abbreviation=P&
    uom=<EPSG-code-of-hPa-UoM>&
  3=http://www.opengis.net/def/crs/ISO/2004/8601
```

In this case we have assumed that each 2D satellite image has been projected to Universal Transverse Mercator (UTM) geographic coordinate system, zone 32 North, represented by the 32632 EPSG code [2]; we specified a generic unidimensional parametrizable CRS called *Linear1D*, which can work as template for any linear 1D axis and which we customized with semantic-specific metadata, namely label, abbreviation and unit of measure of the pressure levels; eventually the widely accepted ISO standard 8601 is used as an opportune way of referring to the temporal dimension [14]. Regarding the pressure axis, the "direction" attribute which is usually required to specify the direct or inverse accordance of the domain direction with the underlying database indexing, is here omitted assuming a default positive direction for these kind of axis. Note that this might not always be true: as an example, images are usually stored in a database taking the upper-left corner as origin, so that the Y pixel indexes increase southwards, thus having a default relative negative direction. It should be underlined how the nature of the discretization of each dimension should be kept separated from its referencing metadata: the same CRS URI would have been applied in case of regularly spaced pressure levels, it is up to the service to properly manage the different cases. The previously defined CCRS URI is enough to handle our 4D dataset: the first two dimensions are defined in the EPSG:32632 definition, first easting then northing as per GML definition[3], then the third dimension is pressure, represented in hPa and finally the temporal axis, supposed to be well-known. The CRS definition are meant to be parsed by the service provider which then can understand the semantic of each single dimension of the multi-dimensional dataset. The GML definition behind the CRS URI of the pressure axis, being it anchored to the same datum of the UTM projected images (WGS84), might be as follows:

```
<gml:EngineeringCRS
  xmlns="http://www.opengis.net/gml"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:gmd="http://www.isotc211.org/2005/gmd"
  xmlns:gco="http://www.isotc211.org/2005/gco"
  xmlns:epsg=
    "urn:x-ogp:spec:schema-xsd:EPSG:0.1:dataset"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  gml:id="Linea1D_example">
  <identifier codeSpace="OGP">
    http://www.opengis.net/def/crs/EPSG/0/5614
  </identifier>
  <name>1D Axis Template</name>
  <scope>General-purpose 1D parametrizable CRS
  </scope>
  <LinearCS gml:id="epsg-cs-6496">
    <identifier codeSpace="OGP">
      http://www.opengis.net/def/cs/EPSG/0/6496
    </identifier>
    <name>Pressure</name>
    <axis>
      <CoordinateSystemAxis
        gml:id="epsg-axis-111"
        gml:uom=
          "http://www.opengis.net/def/uom/EPSG/0/hPa">
        <identifier codeSpace="OGP">
          http://www.opengis.net/def/axis/pressure
        </identifier>
        <axisAbbrev>P</axisAbbrev>
        <axisDirection codeSpace="EPSG">
          positive
        </axisDirection>
      </CoordinateSystemAxis>
    </axis>
  </LinearCS>
  <gml:engineeringDatum>
    <gml:EngineeringDatum gml:id="GeodeticDatum">
      <gml:datumName>WGS_1984</gml:datumName>
      <gml:identifier codeSpace="OGP">
        http://www.opengis.net/def/datum/EPSG/0/6326
      </gml:identifier>
        ...
    </gml:EngineeringDatum>
```

[3]Regarding the axis order defined in a geographic CRS, it might be safer to always force easting coordinates first, as there is no fixed rule in the EPSG database.

```
  </gml:engineeringDatum>
</gml:EngineeringCRS>
```

When receiving the URI of the pressure axis, the CRS resolver under the hood takes the template GML definition of a Linear1D axis and customizes it with the semantic we defined in the URI itself. This same mechanism can be applied for higher dimensional CRSs: to underline the generality of the proposed schema, we might think of the possibility to serve sections of a thoracic computed axial tomography. Similarly in this case the one CRS that would be applied to this collection could be written as:

```
http://www.opengis.net/def/crs?
    authority=OGC&
    version=0.1&
    code=Linear3D&
    axisName1=axial-x&abbreviation1=X&
    axisName1=axial-y&abbreviation1=Y&
    axisName1=vertical-z&abbreviation1=Z
```

In this example we assumed pure numbers representing the position on the images; additionally in the metadata database we might want to define the domain extents in the metadata as the underlying grid dimensions, so as to let the user a 1 to 1 correspondence between the requested areas or volumes of data and the images domain.

In the next subsection we will discuss the utility of this schema when dealing with reprojections in case of unsupported spatial CRSs, such as for planetary imagery.

3.2 Beyond Earth

The URI-oriented approach over CRS definition allows a solution for traditionally *unsupported* reference systems, like in the case of non-geodetic spatial CRSs. On one side, as shown in the previous section, the ability to analyze what is behind a CRS by means of a web resolver would permit the identification of the proper semantics of such datasets; on the other side it could allow for the definitions themselves to be ingested and decoded by cartographic projections libraries[4]. This means that the CRS can be deeply understood by the service, resulting in much more freedom transferred to the user: required subsets would not need to be specified in the native CRS of a planetary dataset, and reprojection of the requested content might be requested as well.

As is the case for geodetic datasets, there is a plenty of different approximating ellipsoids, datums and map projections which can be applied to each single planet: a scientific group might need to handle a variety of observations on a planet, usually each one having its own referencing system. In this specific context the ability to reproject data and/or requested subsets can be of vital importance, since it would allow to spatially synchronize the whole datasets to

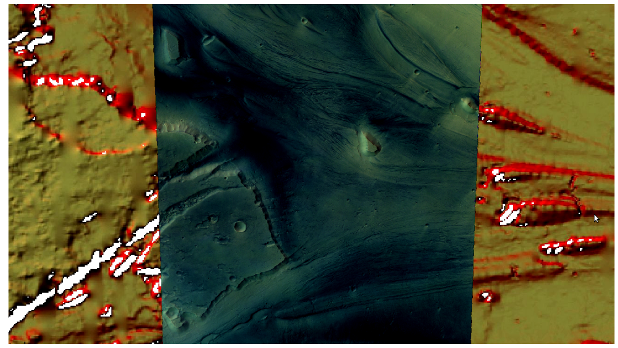

Figure 5: Example of Mars imagery overlays.

a fixed area of interest, thus giving the capability to elaborate location-specific analysis on the available data. Time CRS can certainly be appended to a planetary CRS to correctly manage time series of observations, whereas different spectral bands would still be allowed by the *range* of a collection.

However, with regards to remotely sensed data over non-terrestrial planets or general astronomical objects there might be some additional workload for the service, as still the standards do not contemplate all the possible situations. As an example, the current formats for CRS definitions do not seem to support different latitude types (planetocentric/planetographic): IAU:49910 and IAU:49911 represent two identical equidistant cylindrical projections over Mars, which differ on the latitude type. Their WKT definitions are however identical:

```
PROJCS["Mars_Equidistant_Cylindrical",
    GEOGCS["Mars 2000",
        DATUM["D_Mars_2000",
            SPHEROID["Mars_2000_IAU_IAG",
            3396190.0,169.89444722361179]],
    PRIMEM["Greenwich",0],
    UNIT["Decimal_Degree",0.0174532925199433]],
PROJECTION["Equidistant_Cylindrical"],
PARAMETER["False_Easting",0],
PARAMETER["False_Northing",0],
PARAMETER["Central_Meridian",0],
PARAMETER["Standard_Parallel_1",0],
UNIT["Meter",1]]
```

Still it is up the service to implement the additional processing for coherent reprojection capabilities in such cases.

4. TESTING

The proposed concepts are demonstrated by our Semantic Coordinate Reference System (SECORE) Resolver service. The underlying database is that of EPSG and, hence, knows all its CRS definitions. The service accepts a URL and delivers the corresponding CRS encoded in GML. With the inspiration of the “linked” approach, we design a test crawler to recursively explore properties of the referred CRSs, see Figure 6.

The behavior of the tester is supervised by a combination of

[4]Where GML cannot be decoded, still the resolver's output might be switched by means of an additional key in the URI, otherwise utilities like `gdalsrsinfo` (GDAL >= 1.9.0) could be used on the server side for on-the-fly translations to the much widely known WKT format.

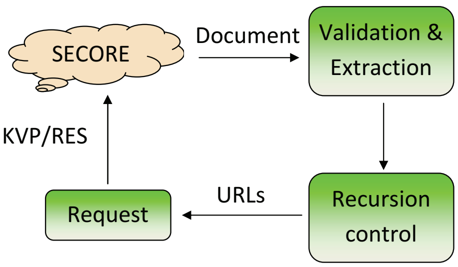

Figure 6: The test recursion procedure for the "linked" resolver.

specified policies. These policies include recursion policies and validation policies. Recursion policies state the conditions to be satisfied to continue the test and the criteria that the tester should use to determine whether the referenced component needs to be checked; the validation policies contain rules that the tester uses to extract the information and validate the referred components. To address the above mentioned scenarios, we define these policies as below: the recursion continues as far as reference components exist, the components are resolved by delivering the corresponding URLs, and the results are validated against the model constrains as specified in ISO 19136 Geographic information – Geographic Markup Language (GML) [13] – and the additional object types as specified in the EPSG Registry model implementation [21]. We implement the test in OGC Compliance Test Language (CTL) [18]. A CTL script is a XML document that describes the testing procedure. The OGC Test, Evaluation, and Measurement (TEAM) Engine runs the script and tests the customer services.

5. CONCLUSIONS

We have presented a naming scheme for CRSs, based on URLs, which allows to express spatial and temporal CRSs of any kind in a uniform, Web-compatible manner. This concept has been established within OGC by consensus among a wide range of geo domains. In particular, the scheme supports the following use cases:

1. Predefined CRSs, such as ISO 8601 for time;
2. Families of predefined CRSs, such as the EPSG list;
3. Compound CRSs, such as the result of an x/t slicing through an x/y/t datacube through a WCS or WCPS operation;
4. Concatenated CRSs, such as when WGS84 is used as a "hub" for transforming between two CRSs where no direct transformation is available;
5. Predefined dimensions (also referred to as "domain axes"), such as latitude.

A registry service, SECORE, has been implemented [17] which allows resolution of CRS definitions identified by CRS URLs. It is based on an XML database and offers resolution services, an equality comparison service, and maintenance of CRS definition. SECORE is part of the open-source "Big Earth Data Analytics" server, `rasdaman` and available from [1]. This implementation has been running in a beta evaluation phase for OGC and is suggested to become the standard OGC resolving service. Among the evaluation scenarios is planetary sciences by establishing a Mars surface and geology service. Hope, therefore, is to pave the way with this CRS Name Type Specification and the resolver for semantic-based machine-machine communication between location-aware services.

6. ACKNOWLEDGEMENT

Work is conducted in the course of the EarthServer project funded through EU FP7-INFRA grant 283610.

7. REFERENCES

[1] www.rasdaman.org, accessed on 2012-jul-02.
[2] EPSG Geodetic Parameter Dataset, 2010. www.epsg-registry.org, accessed on 2012-jul-02.
[3] Guidance Note 7, part 1 – Using the EPSG Geodetic Parameter Dataset, 2011.
[4] Guidance Note 7, part 2 – Coordinate Conversions and Transformation including Formulas, 2012.
[5] P. Baumann. Gml application schema for coverages. *OGC document*, pages 09–146, 2010.
[6] P. Baumann. The OGC web coverage processing service (WCPS) standard. *Geoinformatica*, 14(4):447–479, Oct. 2010.
[7] M. Bock, J. Bohner, O. Conrad, R. Kothe, and A. Ringeler. Saga: System for the automated geoscientific analysis. *Dept. of Physical Geography, Hamburg, Germany. URL http://www. saga-gis. org/en/index. html/(last verified 19 April 2009)*, 2008.
[8] D. Caldeweyher. Open source gis: a grass gis approach. the international series in engineering and computer science. *International Journal of Geographical Information Science*, 25(2):330–331, 2011.
[9] J. de La Beaujardiere. Web map service. *OpenGIS Implementation OGC*, pages 04–024, 2004.
[10] M. Grogan. JSR-223 – Scripting for the JavaTM Platform. Final Draft Specification, version 1.0, Oct. 2006.
[11] ISO 19111:2007: Geographic information – Spatial referencing by coordinates, 2007.
[12] ISO 19123:2005: Geographic information – Schema for coverage geometry and functions, 2005.
[13] ISO 19136:2007: Geographic information âĂŞ Geographic Markup Language (GML), 2007.
[14] ISO 8601:2004: Data elements and interchange formats – Information interchange – Representation of dates and times, 2004.
[15] M. King, W. Menzel, Y. Kaufman, D. Tanré, B. Gao, S. Platnick, S. Ackerman, L. Remer, R. Pincus, and P. Hubanks. Cloud and aerosol properties, precipitable water, and profiles of temperature and water vapor from modis. *Geoscience and Remote Sensing, IEEE Transactions on*, 41(2):442–458, 2003.
[16] Y. Liu and J. Key. Detection and analysis of clear-sky,

low-level atmospheric temperature inversions with modis. *Journal of Atmospheric and Oceanic Technology*, 20(12):1727–1737, 2003.

[17] D. Misev, M. Rusu, and P. Baumann. A semantic resolver for coordinate reference systems. *Web and Wireless Geographical Information Systems*, pages 47–56, 2012.

[18] OGC Compliance Testing Language (CTL) Best Practice, 2011.

[19] OGC Web Coverage Processing Service (WCPS) Language Interface Standard, 2009.

[20] OGC Web Coverage Service (WCS) - Core , 2010.

[21] EPSG Geodetic Parameter Registry - Developer Guide, 2011.

[22] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[23] L. Remer, D. Tanré, and Y. Kaufman. Algorithm for remote sensing of tropospheric aerosol from MODIS: Collection 005. Technical report, National Aeronautics and Space Administration; Goddard Space Flight Center: Greenbelt, MD, 2009.

[24] E. Schmidt, R. Arduini, B. Wielicki, R. Stone, and S. Tsay. Considerations for modeling thin cirrus effects via brightness temperature differences. *Journal of Applied Meteorology*, 34(2):447–459, 1995.

[25] S. Seemann, J. Li, W. Menzel, and L. Gumley. Operational retrieval of atmospheric temperature, moisture, and ozone from modis infrared radiances. *Journal of applied meteorology*, 42(8):1072–1091, 2003.