SIXTH FRAMEWORK PROGRAMME

Project no. **034924**

Project acronym: **VENUS**

Project title: **Virtual ExploratioN of Underwater Sites**

Instrument: SPECIFIC TARGETED RESEARCH OR INNOVATION PROJECT

Thematic Priority [2.5.10] [Access to and preservation of cultural and scientific resources]

D4.3 Archaeological data server + test requests demonstrations

Deliverable No. (use the number indicated on technical annex)	**D4.3**		
Workpackage No.	**WP4**	Workpackage Title	**Mixed Reality Modelling**
Task No.	**4.1**	Task Title	**Modelling and developing an archaeological database engine**
Organisation name of lead contractor for this deliverable	**UEVE**		
Status (F: final; D: draft; RD: revised draft):	**F**		
File Name:	**VENUS_Deliverable43_v10.doc**		
Revision:	**V1.0**		
Due date of deliverable	**January 1st 2008**		
Actual submission date	**February 15st 2008**		
Project start date and duration	**July 1st 2006, 36 Months**		

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	PU
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Author: **David Roussel, UEVE.**
Paul Chapman, UHULL
Luc Long, DRASSM
Pierre Drap, CNRS, Lead Contractor
Control: **Paul Chapman, UHULL**

Modification Log

Version	Date	Author	Description
V0.1	07/01/2008	UEVE	First draft
V0.2	17/01/2008	UEVE	Second draft
V0.3	21/01/2008	UHULL	Third draft
Final (V1.0)	30/01/2008	DRASSM	Final with archaeological needs

Table of contents

1. Introduction

This deliverable follows the first deliverable on the Archaeological Database: **D 4.1 Archaeological database + specification / implementation** report which describes the functionalities provided by the archaeological database for both photogrammetric reconstruction process and virtual environments. D 4.1 also describes the actual database structure. A PDF version describing the D4.1 deliverable can be downloaded at http://piccard.esil.univmed.fr/venus/deliverable/VENUS_Deliverable41_v10.pdf.

The current deliverable focuses on the key requests performed on this archaeological database, but we will first remind the reader about the concerned task goals and actual database structure in order to better understand the goals and needs of the requests presented below.

1.1. Task 4.1: Modelling and developing an archaeological database engine

This task deals with setting up a data model for heterogeneous data such as a Digital Terrain Model (DTM) on one hand and all notes, measures and reconstruction assumptions added to the archaeological site on the other hand. Among theses assumptions, we could find a non exhaustive set of artefacts generic models (such as amphorae, for instance) designed to help archaeologists to create reconstruction assumptions on the site.

This task will also create a first implementation of the above mentioned archaeological database as a prototype in order to test the efficiency of key requests. The relevance of integrating a DTM within a classical relational database also have to be studied in order to choose between a complete integration of the DTM data inside the database or a reference based integration of the DTM within the database with external data. Output indicators: The output indicators for this task consist of two points. The first one is the model of the archaeological database and will be exposed in D4.1. The second one is the integration of this database in the demonstrators which will be illustrated by demonstrations in Deliverable D4.2 & D4.3.

1.2. Archaeological Database structure

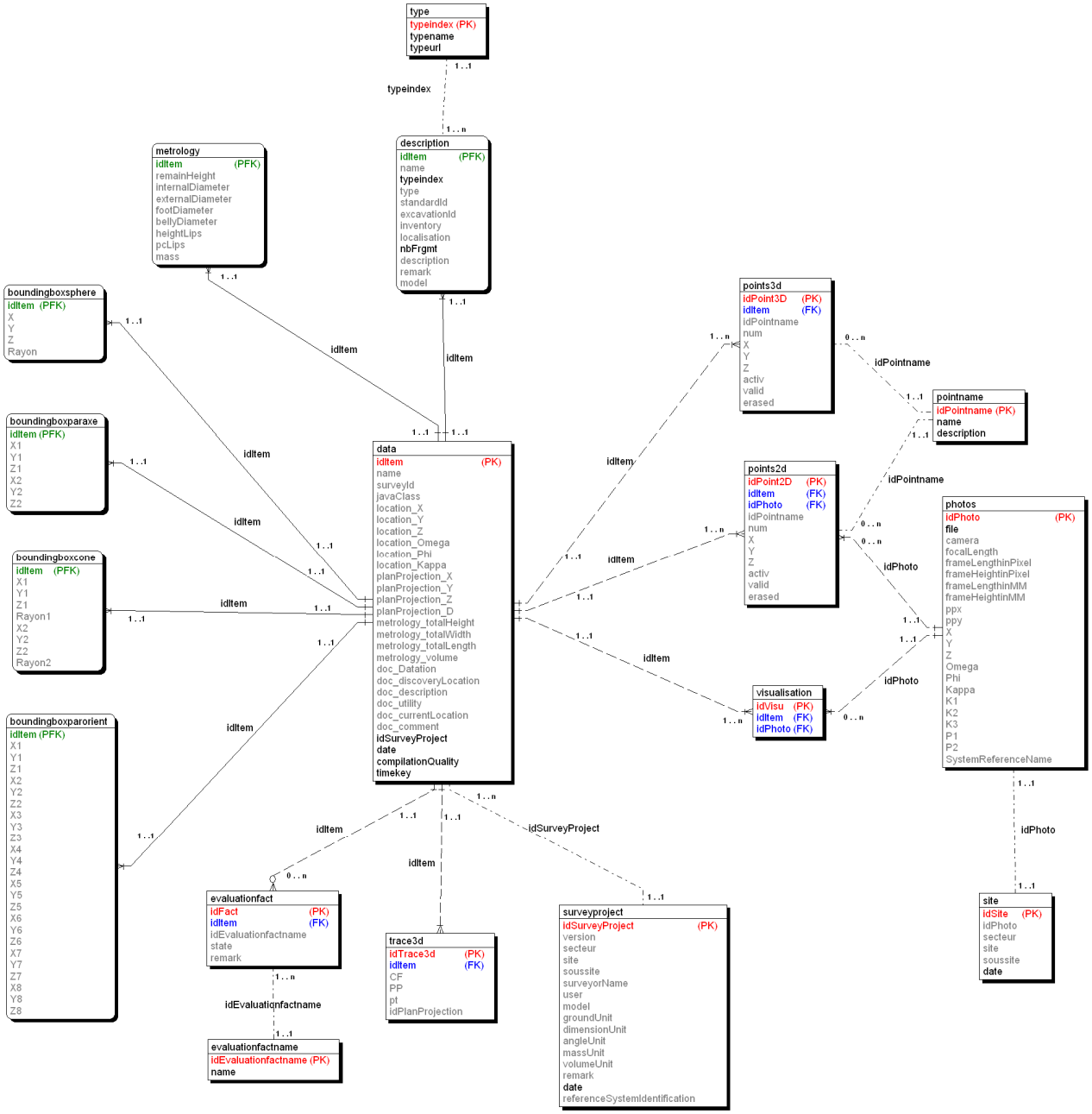

Figure 1: Archaeological Database structure

The database structure is mainly centred around the “data” table which contains reconstructed artefacts parameters. The “data” table also relates to optional parameter tables through a series of one to one relationships. The artefacts parameter of the “data” table are reconstructed by the photogrammetry process by matching 2D points (table “points2d”) on the multiples geolocalised photos (table “photos”) taken on the archaeological site in order to reconstruct 3D points of the seabed and also on the visible artefacts (table “points3d”).

Based on these 3D points the photogrammetry process is able to reconstruct the measured artefacts and store their parameters in the “data” or one of its optional parameters table. For instance location, orientation and sizes of the artefacts are stored in the “data” table, whereas the type of artefact (in our case the type of Amphorae) is stored in the “description” table.

- *Database Amendments*

The various requests presented in this deliverable do not strictly respect the database structure presented above since there are actually several versions of the operating database. All of these versions are not yet filled with data, so in order to present pertinent data request results, we have to introduce a few amendments to the database structure as theses requests were performed on a previous version of the database properly filled with data which we call here the "test database".

The Data Types tables such as "type" related to the "description" table or the "pointname" table related to the "points3d" and "point2d" tables were not present in the test database where the requests were performed.

So the corresponding fields of the missing tables are obtained as follows:

- The "typename" field of the "type" table is obtained through a "subclass" field of the "description" table which is called "descriptionamphore" in the test database.
- The "name" field of the "pointname" table is obtained through the "name" fields of tables "points2d" and "points3d" respectively.

The "metrology" table is described in deliverable 4.1 containing measured and computed value whereas the "metrology" fields of the test database contains only computed values according to standard values observed on previously registered Amphorae types.

These amendments bring only small changes on the requests presented below and these requests will be slightly modified to work properly on the latest version of the database as soon as the lasted version of the database will be filled with data.

2. Archaeological needs

Here are a few examples of requirements provided by archaeologists and questions that archaeologists would like to answer when studying a virtual reconstruction of a site:

- Three-dimensional map […] for observation of the cargo on a screen.
    - Cartography and sections, general dimensions of the "tumulus" (cairn)
- Observation of the different types of amphorae (a lone type?)
- Linking every amphora of the 3D map to a database
    - Click on the amphora to reach the identification sheet
- Number of amphorae layers: *This might be out of scope within Venus framework as it would require excavations that could not be performed by ROVs*.
- Counts of amphorae (complete number of visible amphorae and numbers of amphorae / m^2)
- Proportions of broken amphorae
- General dimensions of the amphorae
- Identification of different sizes of amphorae within the same type
    - Are the amphorae loaded by group of sizes
    - Differentiation by colours of every group of amphorae

3. Database Engine

All of the archaeological databases are powered by a MySQL v5.0.45 server featuring the following storage engines:

Storage Engine	Description
MyISAM	Default engine as of MySQL 3.23 with great performance
MEMORY	Hash based, stored in memory, useful for temporary tables
InnoDB	Supports transactions, row-level locking, and foreign keys
BerkeleyDB	Supports transactions and page-level locking
BLACKHOLE	/dev/null storage engine (anything you write to it disappears)
EXAMPLE	Example storage engine
ARCHIVE	Archive storage engine
CSV	CSV storage engine
ndbcluster	Clustered, fault-tolerant, memory-based tables
FEDERATED	Federated MySQL storage engine
MRG_MYISAM	Collection of identical MyISAM tables
ISAM	Obsolete storage engine

Where the InnoDB support is used to check relational constraints between tables.

Each database has slightly the same structure and weight. Here are the characteristics of the Pianosa test database:

Table	Records	Fields	Type	Collation	Size
boundingboxcone	64	9	InnoDB	*utf8_general_ci*	16.0 KB
boundingboxparaxe	40	7	InnoDB	*utf8_general_ci*	16.0 KB
boundingboxparorient	65	25	InnoDB	*utf8_general_ci*	16.0 KB
boundingboxsphere	64	5	InnoDB	*utf8_general_ci*	16.0 KB
data	65	31	InnoDB	*utf8_general_ci*	128.0 KB
descriptionamphore	65	12	InnoDB	*utf8_general_ci*	64.0 KB
evaluationfact	494	5	InnoDB	*utf8_general_ci*	208.0 KB
metrologyamphore	65	9	InnoDB	*utf8_general_ci*	16.0 KB
photos	293	22	InnoDB	*utf8_general_ci*	352.0 KB
points2d	1,572	11	InnoDB	*utf8_general_ci*	288.0 KB
points3d	1,211	10	InnoDB	*utf8_general_ci*	192.0 KB
site	293	6	InnoDB	*utf8_general_ci*	256.0 KB
surveyproject	65	16	InnoDB	*utf8_general_ci*	96.0 KB
trace3d	0	6	InnoDB	*utf8_general_ci*	32.0 KB
visualisation	130	3	InnoDB	*utf8_general_ci*	48.0 KB
vrml	65	3	InnoDB	*utf8_general_ci*	128.0 KB
x3d	65	3	InnoDB	*utf8_general_ci*	2.0 MB
17 table(s)	**4,616**		***MyISAM***	***utf8_bin***	**3.9 MB**

The archaeological databases are accessible through the VENUS piccard server: mysql://piccard.esil.univmed.fr:3306/PianosaXX where XX indicates the database version.

4. Database queries

4.1. Inventory type requests

The first kind of request to be performed on the database concerns the need to have an inventory of the archaeological site in terms of artefact types and population.

The number of items or artefacts registered on the site could be obtained by counting the number of elements in the "data" table or any other table maintaining a one to one relationship with the data table.

Note: assigning symbolic names to tables in the `FROM` clause such as "`FROM`\`descriptionamphore\` S" is not mandatory in common SQL language but helps keep request fields shorter by prefixing fields by the symbolic name (`S.subClass` for instance) rather than by complete table name (`descriptionamphore.subClass`).

- *Request: Items counts*

```
-- Items count in the database
SELECT
      COUNT(*) AS 'Items count'
FROM
      Data
```

Request result

Items count
65

Then we would like to find out the different kinds of artefacts registered

- *Request: Items types*

```
-- Items types in the database
SELECT DISTINCT
      S.subClass AS 'Type'
FROM
      `descriptionamphore` S
```

Request result

Type
Dressel2_4_longue
Dressel2_4_Courte
Dressel_20
Beltran_2B
Gauloise_3
Amphore
Haltern_70
Pascual_1
Dressel7_11

Where "Amphore" indicates an unrecognized type of amphorae.

- *Request: Item types count*

```
-- Items count types in the database
SELECT
      count(DISTINCT S.subClass) AS 'Type count'
FROM
      `descriptionamphore` S
```

Request result

Type count
9

The next request set up a brief inventory of the site by showing only items type and fragment status of the item.

- *Request: Items, types and fragment status*

```
-- Items & types in the database
SELECT
	D.idItem AS 'Item',
	S.subClass AS 'Type',
	S.name AS 'Frag Type'
FROM
	`data` D,
	`descriptionamphore` S
WHERE
	D.idItem = S.idItem
ORDER BY
	D.idItem
```

Request result

Item	Type	Frag Type
1	Dressel2_4_longue	Full
2	Dressel2_4_Courte	Full
3	Dressel_20	Full
4	Beltran_2B	Full
5	Gauloise_3	Full
6	Amphore	Full
7	Haltern_70	Full
8	Haltern_70	Full
9	Pascual_1	Full
…	…	…
40	Dressel7_11	Full
41	Dressel2_4_Courte	Foot
42	Dressel2_4_longue	Foot
43	Dressel7_11	Belly
…	…	…
65	Dressel2_4_longue	Belly

"Full", "Foot" and "Belly" have been roughly translated in here from "Entiere", "Fond" and "Panse" which indicates if the artefact is fully preserved or if it is a fragment of a specific part of an amphora. The request above could then be specialized in order to build separate inventories for full artefacts and fragments:

- *Requests: Separate items and fragments lists*

```
-- Full Amphorae
SELECT
	D.idItem AS 'Item',
	S.subClass AS 'Type'
FROM
	`data` D,
	`descriptionamphore` S
WHERE
	D.idItem = S.idItem
	AND S.name = "Entiere"
ORDER BY
	D.idItem
```

```
-- Fragments
SELECT
	D.idItem AS 'Item',
	S.subClass AS 'Type',
	S.name AS 'Frag Type'
FROM
	`data` D,
	`descriptionamphore` S
WHERE
	D.idItem = S.idItem
	AND S.name != "Entiere"
ORDER BY
	D.idItem
```

```
-- Full Amphorae count
SELECT
	COUNT(*) AS 'Count Entiere'
FROM
	data D
WHERE
	D.name = "Entiere"
```

```
-- Fragment count
SELECT
	COUNT(*) AS 'Count Fragments'
FROM
	data D
WHERE
	D.name != "Entiere"
```

Requests results:

Count Entiere
40

Count Fragments
25

We could then build separate inventories for full artefacts and fragments by type of artefacts with counts and relative percentage:

- *Requests: Types, counts and percentage of items*

```
-- Types, counts and percentage of
full artefacts
SELECT
      S.subClass AS 'Type',
      COUNT(*) AS 'Count',
      100 * COUNT(*)/(SELECT
      COUNT(*) FROM
      `descriptionamphore` S WHERE
      S.name="Entiere") AS 'Percent'
FROM
      `descriptionamphore` S
WHERE
      S.name = "Entiere"
GROUP BY
      S.subClass
```

```
-- Types, counts and percentage of
fragments
SELECT
      S.subClass AS 'Type',
      COUNT(*) AS 'Count',
      100 * COUNT(*)/(SELECT
      COUNT(*) FROM
      `descriptionamphore` S WHERE
      S.name!="Entiere") AS
      'Percent'
FROM
      `descriptionamphore` S
WHERE
      S.name != "Entiere"
GROUP BY
      S.subClass
```

Both requests uses a nested request in order to count all full artefacts against all full artefacts of a specific type obtained by the `S.subClass` grouping.

Requests results:

Full artefacts		
Type	Count	Percent
Amphore	1	2.5
Beltran_2B	1	2.5
Dressel2_4_Courte	12	30
Dressel2_4_longue	16	40
Dressel7_11	5	12.5
Dressel_20	1	2.5
Gauloise_3	1	2.5
Haltern_70	2	5.0
Pascual_1	1	2.5

Fragments		
Type	Count	Percent
Amphore	1	4.0
Dressel2_4_Courte	11	44.0
Dressel2_4_longue	9	36.0
Dressel7_11	3	12.0
Dressel_20	1	4.0

4.2. Statistics requests

The database could also be used to build statistics on the various dimensions of the different kinds of artefacts: For instance Height and Diameters of full amphorae (Width and Length are equals for amphorae on the Pianosa site, but will be used on the new Sesimbra archaeological site composed of bricks and tiles).

- *Request: Statistics on Full Amphorae sizes by types*

```
SELECT
	S.subClass AS 'Type',
	COUNT(ALL S.idItem) AS 'Count',
	AVG(D.metrology_totalHeight)*100 AS 'Mean Height',
	STD(D.metrology_totalHeight)*100 AS 'Stddev Height',
	MIN(D.metrology_totalHeight)*100 AS 'Min Height',
	MAX(D.metrology_totalHeight)*100 AS 'Max Height',
	AVG(D.metrology_totalWidth)*100 AS 'Mean Diameter',
	STD(D.metrology_totalWidth)*100 AS 'Stddev Diameter',
	MIN(D.metrology_totalWidth)*100 AS 'Min Diameter',
	MAX(D.metrology_totalWidth)*100 AS 'Max Diameter'
FROM
	`data` D,
	`descriptionamphore` S
WHERE
	D.idItem = S.idItem
	AND D.name = "Entiere"
GROUP BY
	S.subClass
ORDER BY
	AVG(D.metrology_totalHeight)
```

Requests results:

Type	Count	Mean Height	Stddev Height	Min Height	Max Height	Mean Diameter	Stddev Diameter	Min Diameter	Max Diameter
Dressel_20	1	66.60	0.00	66.60	66.60	53.42	0.00	53.42	53.42
Gauloise_3	1	76.30	0.00	76.30	76.30	53.70	0.00	53.70	53.70
Haltern_70	2	79.30	0.00	79.30	79.30	33.42	0.68	32.74	34.11
Dressel7_11	5	83.84	3.63	76.60	86.06	33.62	5.25	25.36	40.54
Dressel2_4_Courte	12	85.20	0.68	85.00	87.45	31.41	4.59	27.33	41.45
Amphore	1	96.89	0.00	96.89	96.89	39.87	0.00	39.87	39.87
Pascual_1	1	98.00	0.00	98.00	98.00	36.02	0.00	36.02	36.02
Dressel2_4_longue	16	102.49	4.93	90.43	105.00	28.91	1.62	25.71	30.83
Beltran_2B	1	112.60	0.00	112.60	112.60	26.69	0.00	26.69	26.69

Where heights and diameters are expressed in centimetres.

The values obtained from the previous request could then be compared to the standard dimensions values contained in the “metrology” table:

- *Request: Statistics on full artefacts heights compared to standard values*

```
SELECT
	S.subClass AS 'Type',
	COUNT(ALL S.idItem) AS 'Count',
	AVG(D.metrology_totalHeight)*100 AS 'Mean Height',
	STD(D.metrology_totalHeight)*100 AS 'Stddev Height',
	MIN(D.metrology_totalHeight)*100 AS 'Min Height',
	MAX(D.metrology_totalHeight)*100 AS 'Max Height',
```

```
	AVG(M.remainHeight)*100 AS 'Standard Height',
	ABS((AVG(D.metrology_totalHeight)-AVG(M.remainHeight)) /
	AVG(M.remainHeight))*100 AS '% difference'
FROM
	`data` D,
	`descriptionamphore` S,
	`metrologyamphore` M
WHERE
	D.idItem = S.idItem
	AND D.idItem = M.idItem
	AND D.name = "Entiere"
GROUP BY
	S.subClass
HAVING
	COUNT(ALL S.idItem) > 1
```

Requests results:

Type	Count	Mean Height	Stddev Height	Min Height	Max Height	Standard Height	% difference
Dressel2_4_Courte	12	85.20	0.68	85.00	87.45	85.00	0.24
Dressel2_4_longue	16	102.49	4.93	90.43	105.00	105.00	2.39
Dressel7_11	5	83.84	3.63	76.60	86.06	85.52	1.96
Haltern_70	2	79.30	0.00	79.30	79.30	79.30	0.00

Where heights and diameters are expressed in centimetres and singletons have been avoided by the `HAVING COUNT(ALL S.idItem) > 1` clause.
We could see that mean height of non singleton populations are quite similar to the standard height values for these artefacts types.

Similar Requests can be performed on other measures such as Diameter by replacing `D.metrology_totalHeight` by `D.metrology_totalWidth` and `M.remainHeight` by `M.bellyDiameter`.

- *Request: Statistics on full artefacts diameters compared to standard values*

```
SELECT
	S.subClass AS 'Type',
	COUNT(ALL S.idItem) AS 'Count',
	AVG(D.metrology_totalWidth)*100 AS 'Mean Diameter',
	STD(D.metrology_totalWidth)*100 AS 'Stddev Diameter',
	MIN(D.metrology_totalWidth)*100 AS 'Min Diameter',
	MAX(D.metrology_totalWidth)*100 AS 'Max Diameter',
	AVG(M.bellyDiameter)*100 AS 'Standard Diameter',
	ABS((AVG(D.metrology_totalWidth)-AVG(M.bellyDiameter)) /
	AVG(M.bellyDiameter))*100 AS '% difference'
FROM
	`data` D,
	`descriptionamphore` S,
	`metrologyamphore` M
WHERE
	D.idItem = S.idItem
	AND D.idItem = M.idItem
	AND D.name = "Entiere"
GROUP BY
	S.subClass
HAVING
	COUNT(ALL S.idItem) > 1
```

Requests results:

Type	Count	Mean Diameter	Stddev Diameter	Min Diameter	Max Diameter	Standard Diameter	% difference
Dressel2_4_Courte	12	31.41	4.59	27.33	41.45	34.50	8.97
Dressel2_4_longue	16	28.91	1.62	25.71	30.83	31.00	6.76
Dressel7_11	5	33.62	5.25	25.36	40.54	38.00	11.53
Haltern_70	2	33.42	0.68	32.74	34.11	29.70	12.54

4.3. Virtual Environments requests

The first requests performed during the Virtual Environment launching concerns loading the artefacts parameters (full items as well as fragments) followed by loading fragments 3D points.

First we retrieve the artefacts parameters:

- *Request: Virtual Environment artefacts loading*

```
-- Loads Artefacts parameters : i.e. Amphorae (Full AND Broken)
--    Item number       D.idItem or S.idItem
--    Type              S.subClass
--    Fragment status   D.name
--    Location          D.location_X, D.location_Y, D.location_Z
--    Orientation       D.location_Omega, D.location_Phi, D.location_Kappa
--    Dimensions        D.metrology_totalHeight, D.metrology_totalWidth
--    Remarks                S.remark
SELECT
      D.idItem AS 'Item',
      S.subClass AS 'Type',
      D.name AS 'Frag',
      D.location_X AS 'X',
      D.location_Y AS 'Y',
      D.location_Z AS 'Z',
      D.location_Omega AS 'rX',
      D.location_Phi AS 'rY',
      D.location_Kappa AS 'rZ',
      D.metrology_totalHeight AS 'Height',
      D.metrology_totalWidth AS 'Width',
      S.remark AS 'Comment'
FROM
      `data` D,
      `descriptionamphore` S
WHERE
      D.idItem = S.idItem
```

Requests results:

Item	Type	Frag	X	Y	Z	rX	rY	rZ	Height	Width	Comment
1	Dressel2_4_longue	Entiere	808.954	906.568	-32.686	4.647	0.439	4.374	1.011	0.295	markers 10 15
2	Dressel2_4_Courte	Entiere	808.596	908.776	-32.758	1.590	0.925	3.185	0.850	0.335	cassee sous epaules
3	Dressel_20	Entiere	796.523	912.297	-32.890	4.022	2.530	4.572	0.666	0.534	cassee demi panse, marker 1
4	Beltran_2B	Entiere	800.795	907.893	-32.458	4.370	0.664	-1.418	1.126	0.267	cassee soue epaules, marker 13
5	Gauloise_3	Entiere	801.260	907.041	-33.330	3.384	2.978	1.353	0.763	0.537	dans le sable, marker 13
...	...	...	...	...	...	...	...	...	...	...	...
40	Dressel7_11	Entiere	799.757	913.282	-32.792	4.566	0.673	4.171	0.766	0.405	marker 2
41	Dressel2_4_Courte	Fond	802.477	907.088	-32.723	3.004	3.029	3.197	0.000	0.000	markers 13 14
42	Dressel2_4_longue	Fond	802.570	902.128	-32.548	3.099	3.684	1.949	0.000	0.000	no numero, externe
43	Dressel7_11	Panse	799.586	913.480	-32.667	3.059	3.084	4.321	0.000	0.000	marker 2
...	...	...	...	...	...	...	...	...	...	...	...
65	Dressel2_4_longue	Panse	797.796	908.886	-32.479	3.205	3.469	2.454	0.000	0.000	entre amph 28r et regle rouge

Then we load the remaining 3D points of each fragments: The fragments are not yet reconstructed, so we only could load the corresponding 3D points.

- *Request: Virtual Environment broken artefacts 3D points loading*

```
-- Loads Broken Artefacts 3D points
--    Item number       D.idItem or S.idItem
--    Type              S.subClass
--    Fragment Type     D.name
--    3D point number   P.idPoint3D
--    3D point coords   P.X, P.Y, P.Z
--    3D point type     P.name
SELECT
      D.idItem AS 'Item',
      S.subClass AS 'Type',
      D.name AS 'Fragment',
      P.idPoint3D AS 'idPoint3D',
      P.X AS 'X',
      P.Y AS 'Y',
      P.Z AS 'Z',
      P.name AS 'type'
FROM
      `data` D,
      `descriptionamphore` S,
      `points3d` P
WHERE
      D.idItem = S.idItem
      AND D.idItem = P.idItem
      AND D.name != "Entiere"
```

Requests results:

Item	Type	Fragment	idPoint3D	X	Y	Z	Ftype
41	Dressel2_4_Courte	Fond	2867	802.664	907.067	-32.624	F_PER
41	Dressel2_4_Courte	Fond	2868	802.626	906.976	-32.649	F_PER
41	Dressel2_4_Courte	Fond	2869	802.537	906.889	-32.671	F_PER
…	…	…	…	…	…	…	…
42	Dressel2_4_longue	Fond	2880	802.661	902.342	-32.536	F_PER
42	Dressel2_4_longue	Fond	2881	802.665	902.299	-32.546	F_PER
42	Dressel2_4_longue	Fond	2882	802.716	902.248	-32.536	F_PER
42	Dressel2_4_longue	Fond	2896	802.57	902.128	-32.548	F_FND
…	…	…	…	…	…	…	…
43	Dressel7_11	Panse	2897	799.694	913.252	-32.808	F_PER
43	Dressel7_11	Panse	2898	799.664	913.259	-32.757	F_PER
43	Dressel7_11	Panse	2899	799.607	913.251	-32.751	F_PER
43	Dressel7_11	Panse	2900	799.587	913.232	-32.79	F_PER
43	Dressel7_11	Panse	2901	799.56	913.278	-32.779	F_PER
43	Dressel7_11	Panse	2902	799.524	913.32	-32.811	F_PER
…	…	…	…	…	…	…	…

Once these data are loaded the seabed can then be populated with the corresponding artefacts.

Inventory type requests as well as global statistics requests results are not well fitted to a virtual environment. Indeed, a virtual environment is not the most promising way to display large table results.

Figure 2: Amphorae type statistics on page 17 shows an early attempt to display statistics on a specific type of artefact (in this particular case Dressel 2/4 Long Amphorae).

Figure 2: Amphorae type statistics

So Tabular statistics are not easily displayed in a Virtual Environment. However, request results are easily displayed graphically by selecting and highlighting artefacts corresponding to a specific request result: for instance, selecting full artefacts of a specific type, height, diameter or any other criteria resulting in a list of items easily highlight able rather than numerical values.

So the generic form of the requests performed from within the Virtual Environment should take the following form:

- *Request: Virtual Environment generic selection request form*

```
SELECT
        D.idItem
FROM
        `data` D,
        <Any other tables>
WHERE
        D.idItem = <An other table>.idItem
        AND D.idItem = <Any other table>.idItem
        AND <Other Selection criteria>
        AND D.name = "Entiere"
ORDER BY
        D.idItem ASC
```

These kinds of selection requests are generally associated with an equivalent count request in order to display the number of selected items:

- *Request: Virtual Environment generic count request form*

```
SELECT
       COUNT(*)
FROM
       `data` D,
       <Any other tables>
WHERE
       D.idItem = <An other table>.idItem
       AND D.idItem = <Any other table>.idItem
       AND <Other Selection criteria>
       AND D.name = "Entiere"
```

For instance, the selection and count of all “Dressel 2/4 Long” Amphorae are performed with the following requests:

```
-- All Full Dressel 2/4 Long selection
SELECT
       D.idItem
FROM
       `data` D,
       `descriptionamphore` S
WHERE
       D.idItem = S.idItem
       AND S.subClass = "Dressel2_4_longue"
       AND D.name = "Entiere"
ORDER BY
       D.idItem ASC
```

```
-- All Full Dressel 2/4 Long Count
SELECT
       COUNT(*)
FROM
       `data` D,
       `descriptionamphore` S
WHERE
       D.idItem = S.idItem
       AND S.subClass = "Dressel2_4_longue"
       AND D.name = "Entiere"
```

Requests results: The results are in this case displayed graphically by highlighting the corresponding artefacts and displaying the count of selected items.

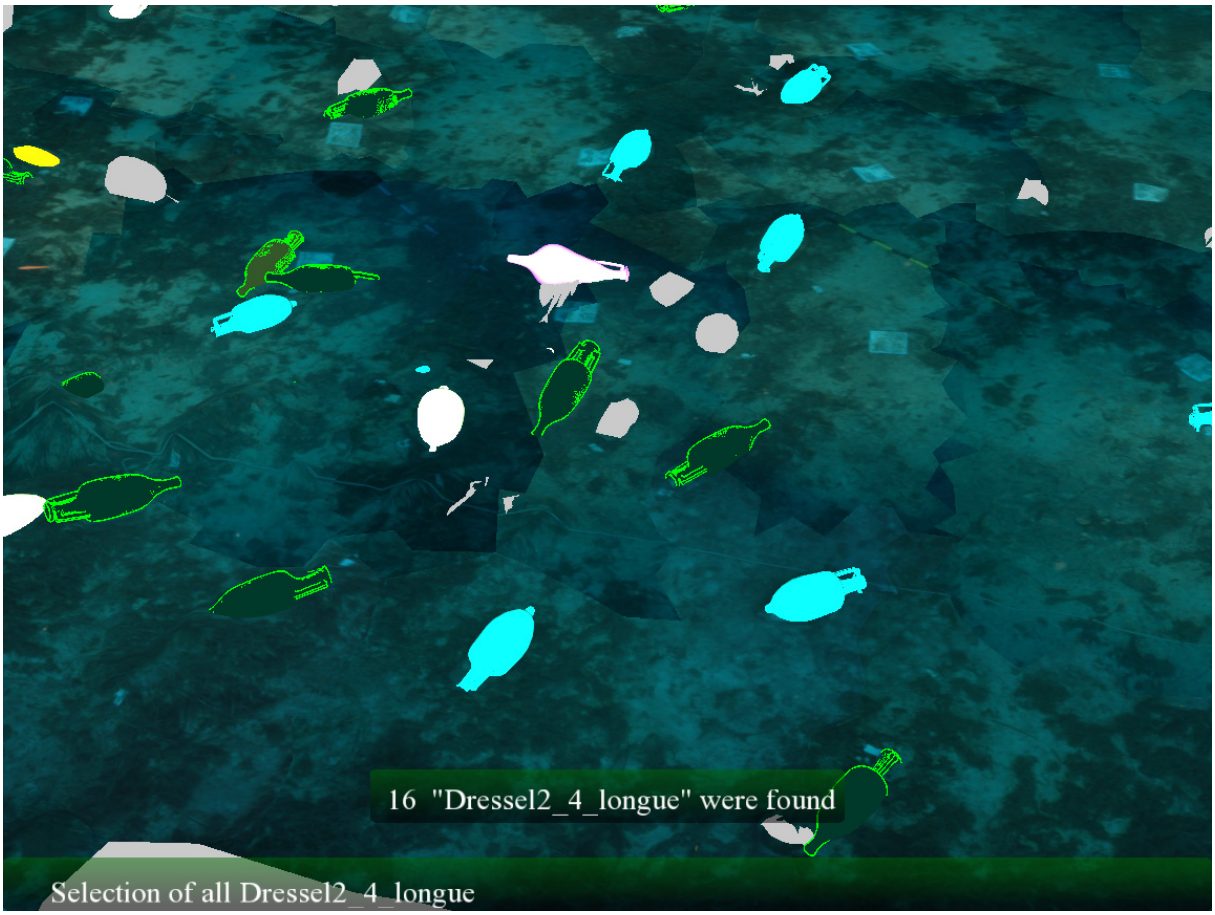

Figure 3: Artefacts selection request results example

Another common way to perform requests from the Virtual Environment is to use the parameters of a selected item to perform comparison requests based on the selected item's height, diameter or any other parameter.

For instance, if the currently selected artefact is a Dressel 2/4 Long and has a height contained within 90 to 100 cm, we could select artefacts of all types featuring similar heights or restrict the search to artefacts of the same type:

- *Request: Artefacts Selection featuring a height within a range*

```
SELECT
	D.idItem AS 'Item',
	S.subClass AS 'Type',
	D.metrology_totalHeight AS 'Height'
FROM
	`data` D,
	`descriptionamphore` S
WHERE
	D.idItem = S.idItem
	AND D.name = "Entiere"
	AND D.metrology_totalHeight BETWEEN 0.90 AND 1.0
ORDER BY
	D.idItem ASC
```

Where "`0.90 AND 1.0`" are derived from the currently selected artefact height. The "Item" column is then used to highlight the corresponding artefacts in the Virtual Environment.

Requests results:

Item	Type	Height
6	Amphore	0.969
9	Pascual_1	0.980
22	Dressel2_4_longue	0.977
29	Dressel2_4_longue	0.906
39	Dressel2_4_longue	0.904

We could also restrict the search to Dressel 2/4 Long type by adding a "`AND S.subClass = "Dressel2_4_longue"`" criteria to the **`WHERE`** clause and obtain only items number 22, 29 and 39.

Assume that the currently selected artefact is item 22 (Dressel 2/4 Long) featuring a height of 97.7 cm, we could try to find the count of similar artefact types with the following request displaying the types and occurrences in these types satisfying similar heights, followed by the mean difference in heights:

- *Request: Artefact Type selection featuring the closest height*

```
SELECT
	S.subClass AS 'Type',
	COUNT(S.subClass) AS 'Occurrences',
	ABS(AVG(D.metrology_totalHeight)-0.976878897) AS 'Diff Height'
FROM
	`data` D,
	`descriptionamphore` S
WHERE
	D.idItem = S.idItem
	AND D.name = "Entiere"
	AND D.metrology_totalHeight BETWEEN 0.90 AND 1.0
GROUP BY
	S.subClass
ORDER BY
	COUNT(S.subClass) DESC
```

Where `0.976878897` is the height of the currently selected artefact

Requests results:

Type	Occurrences	Diff Height
Dressel2_4_longue	3	0.0479
Pascual_1	1	0.0031
Amphore	1	0.0080

In this particular case the results stays elusive as we have a higher occurrences of Dressel 2/4 Long but with a larger difference in heights compared to the two other lower occurrences.

Suppose that the currently selected item is #37, a Dressel 2/4 Long amphora located at X=798.55 & Y=905.254. The following request finds the artefacts of any type located around the current location within a range of 2 meters away:

- *Request: Selection of any type artefacts within a distance range*

```
-- Request 091 : Selection of any type artefacts within a distance range of
-- 2 meters away in XY plane from a reference point
-- reference point X=798.55 Y=905.254 belongs to artefact #37
SELECT
	D.idItem AS 'Item',
	S.subClass AS 'Type',
	D.location_X AS 'X',
	D.location_Y AS 'Y',
	SQRT(((D.location_X-798.55)*(D.location_X-798.55)) + ((D.location_Y-
905.254)*(D.location_Y-905.254))) AS 'Distance'
FROM
	`data` D,
	`descriptionamphore` S
WHERE
	D.idItem = S.idItem
	AND D.idItem != 37 -- because it's the currently selected item
	AND D.name = "Entiere"
	AND SQRT(((D.location_X-798.55)*(D.location_X-798.55)) + ((D.location_Y-
905.254)*(D.location_Y-905.254))) < 2.0
```

This request could also be limited to Dressel 2/4 Long Amphorae types by adding an "`AND S.subClass = "Dressel2_4_longue"`" criteria to the **WHERE** clause.

Requests results:

Item	Type	X	Y	Distance
15	Dressel2_4_longue	799.373	906.611	1.587
16	Dressel2_4_longue	797.212	905.936	1.502
17	Dressel2_4_Courte	797.202	904.261	1.674
35	Dressel7_11	797.193	904.694	1.468
36	Dressel2_4_longue	796.813	905.770	1.812
38	Dressel2_4_longue	798.840	903.894	1.391

Suppose now that the currently selected item is item #1, a Dressel 2/4 Long Amphora featuring an orientation of `4.647` radians along the X axis, along the Y axis and along the Z axis. The following request searches for full artefacts of the same type and featuring similar orientations within an range of $\frac{\pi}{5}$:

- *Request: Dressel2_4_longue selection having similar orientations*

```
-- Request 080 : Dressel2_4_longue selection having similar orientations
-- as item #1 within a PI/5 range
SELECT
	D.idItem,
	D.location_Omega,
	D.location_Phi,
	D.location_Kappa
FROM
	`data` D,
	`descriptionamphore` S
WHERE
	D.idItem = S.idItem
	AND D.name = "Entiere"
	AND S.subClass = "Dressel2_4_longue"
	AND (D.location_Omega BETWEEN 4.64701721855162-0.628
		AND 4.64701721855162+0.628)
```

```
        AND (D.location_Kappa BETWEEN 4.37421140689105-0.628
              AND 4.37421140689105+0.628)
        AND (D.location_Phi BETWEEN 0.439075837782815-0.628
              AND 0.439075837782815+0.628)
ORDER BY
        D.idItem ASC
```

The right way to compute similar orientations would be to compute a cross or dot product between vectors since the norm of the cross product of unit vectors is proportional to the sinus of the angle between vectors (respectively the dot product is proportional to the cosinus of this angle), but this computation would imply to apply theses rotations along X, Y and Z axis to unit vectors which would require matrix computation and would be really nontrivial to translate into a single formula in SQL syntax.

Requests results:

idItem	location_Omega	location_Phi	location_Kappa
1	4.647	0.439	4.374
16	4.608	0.292	3.959
34	4.640	0.480	3.796

Similar orientation requests are not very relevant on the Pianosa archaeological site, but it would become extremely relevant on the new Sesimbra site since, this site features large heaps of roof tiles, hence featuring similar orientation and grouped locations.

We could notice that even if the two last requests above are adequate, they are a bit cumbersome since mathematical formulas are not easily described in SQL syntax. On the other hand, such requests could have been easily computed directly within the Virtual Environment since it mainly manipulates 3D points and vectors and looking for the closest artefacts or similar orientations would be an easy task to perform by computing distances and cross products (As a matter of fact cross product is not defined in SQL and the equivalent formulation would require lines of code because SQL statements do not include variables for instance).

This observation raises the problem of querying the Virtual Environment directly rather than querying the archaeological database. The answer relies in the data loaded by the Virtual Environment from the database (see the two first requests of the virtual Environment on page 15). If the request concerns the data already loaded it would be faster and easier to perform the request directly within the Virtual Environment.

4.4. Photogrammetry post-processing

The requests related to photogrammetry are mainly used to retrieve artefacts based on selected photos, or photos based on selected items and/or more generally relationships between photos, artefacts, 2D and 3D points.

Concerning the relationships between the artefacts and the 3D points, the following request display the list of 3D points used to reconstruct this artefact:

- *Request: 3D points lists of a specific item*

```
SELECT
	P.idPoint3D AS 'idPoint3D',
	P.name AS 'type',
	P.num AS 'num',
	P.X AS 'X',
	P.Y AS 'Y',
	P.Z AS 'Z'
FROM
	`data` D,
	`points3d` P
WHERE
	D.idItem = 1
	AND D.idItem = P.idItem
ORDER BY
	P.name
```

Requests results:

idPoint3D	type	num	X	Y	Z
2077	A_AN1	5	809.738	906.980	-32.689
2078	A_AN2	5	809.832	906.797	-32.718
2080	A_CMX	6	809.359	906.859	-32.608
…	…	…	…	…	…
2072	A_COL	1	809.866	907.001	-32.716
…	…	…	…	…	…
2087	A_CUL	12	808.954	906.568	-32.686
2095	CenterCircleBelly	1	809.330	906.728	-32.673
…	…	…	…	…	…
2092	CenterCircleLips	1	809.896	906.929	-32.746
…	…	…	…	…	…

Where the type columns contains the area of the artefact where the points was picked.

The following request counts the number of 3D points measured on every artefacts:

- *Request: Counts of 3D points measured on each artefacts*

```
SELECT
	D.idItem AS 'Item',
	S.subClass AS 'Type',
	D.name AS 'Frag',
	COUNT(D.idItem) AS 'Nb Points3D'
FROM
	`data` D,
	`descriptionamphore` S,
	`points3d` P
```

```
WHERE
      D.idItem = S.idItem
      AND D.idItem = P.idItem
GROUP BY
      D.idItem
```

This request could be restricted to full artefacts by adding a "`AND D.name = "Entiere"`" criteria to the `WHERE` clause, or it could be restricted to a selected item number by adding a "`AND D.idItem =` <*selected item index*>" criteria.

Requests results:

Item	Type	Frag	Nb Points3D
1	Dressel2_4_longue	Entiere	32
2	Dressel2_4_Courte	Entiere	28
3	Dressel_20	Entiere	23
4	Beltran_2B	Entiere	25
5	Gauloise_3	Entiere	18
6	Amphore	Entiere	26
…	…	…	…
40	Dressel7_11	Entiere	30
41	Dressel2_4_Courte	Fond	13
42	Dressel2_4_longue	Fond	17
43	Dressel7_11	Panse	24
…	…	…	…
65	Dressel2_4_longue	Panse	9

Concerning the relationships between the artefacts and the photos of the site, we can choose between two paths (see Figure 1: Archaeological Database structure on page 5): the first one can use the "points2d" table registering 2D points used to produce 3D points. The second one can use the "visualisation" table registering which artefact is seen in which photo or which photo sees which artefacts. Until now these two paths deliver the same results as the "visualisation" table only contains relationship between photos used to measure items and measured items, further developments are required to extend the visualisation table data to all the photos including those not used to measure any artefact but which sees some artefacts.

- *Requests: List of photos by artefacts*

```
-- Request 110 : List of Photos used
-- to measure the different
-- artefacts (using
-- visualisation table)
SELECT
      D.idItem AS 'Item',
      P.idPhoto AS 'idPhoto',
      P.file AS 'URL'
FROM
      `data` D,
      `visualisation` V,
      `photos` P
WHERE
      D.idItem = V.idItem
      AND V.idPhoto = P.idPhoto
```

```
-- Request 110 : List of Photos used
-- to measure the different
-- artefacts (using points2d table)

SELECT DISTINCT
      D.idItem AS 'Item',
      P.idPhoto AS 'idPhoto',
      P.file AS 'URL'
FROM
      `data` D,
      `points2d` P2,
      `photos` P
WHERE
      D.idItem = P2.idItem
      AND P2.idPhoto = P.idPhoto
```

Requests results:

idItem	idPhoto	file
1	81	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_4010.JPG
1	82	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_4011.JPG
2	81	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_4010.JPG
2	80	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_4009.JPG
3	1	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_4051.JPG
3	2	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_4052.JPG
…	…	…
65	51	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_3980.JPG
65	53	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_3982.JPG

Another way to use this relationship is to display the measured or visible artefacts by photos. If the relationship uses the `points2d` table, it would only concerns measured items whereas if we use the `visualisation` table it would then concern the visible artefacts:

- *Request: Measured artefacts by photos*

```
-- Request 112 : measured artefacts by photos (using points2d table)
SELECT DISTINCT
	D.idItem AS 'Item',
	P.idPhoto AS 'idPhoto',
	P.file AS 'URL'
FROM
	`data` D,
	`points2d` P2,
	`photos` P
WHERE
	D.idItem = P2.idItem
	AND P2.idPhoto = P.idPhoto
ORDER BY
	P.idPhoto
```

Requests results:

idItem	idPhoto	file
3	1	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_4051.JPG
10	2	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_4052.JPG
3	2	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_4052.JPG
10	3	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_4053.JPG
15	6	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_3933.JPG
14	7	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_3934.JPG
15	7	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_3934.JPG
46	8	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_3935.JPG
27	8	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_3935.JPG
54	8	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_3935.JPG
5	8	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_3935.JPG
14	8	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_3935.JPG
4	8	http://piccard.esil.univmed.fr/venus-archives/pianosa/diver/DSC_3935.JPG
…	…	…

5. Conclusion

The archaeological server is currently up and running and supports several versions of the Pianosa site and the Sesimbra site is currently submitted to the photogrammetric reconstruction process in order to produce the measured artefacts (See Deliverable 3.3: Knowledge based photogram metric software interface 2).

The various requests presented in this deliverable are currently integrated within the Virtual Environment demonstrator engine considering that requests concerning all data already loaded (i.e. locations, orientations and dimensions of the artefacts) should be preferably performed within the Virtual Environment itself rather than querying the database with complicated mathematical requests.