Background

There have been several buildingSMART and related projects aimed at the capture of architectural programming information to support both the architectural programming effort, delivery of standard Request for Proposal documents to more clearly communicate owners’ requirements, and to perform the automated assessment of spatial compliance of later design documents. These projects include the Portfolio and Asset management – Performance Requirements (PAMPeR), International Alliance for Interoperability’s AR-5 Project, the buildingSMART international Room Data Sheet “aquarium” project, the United States General Services Administration project (Concept Design BIM 2010 – Spatial Program Validation), the buildingSMART alliance Spatial Compliance information exchange (SCie, pronounced “ski”) project, and the Norwegian effort “IDM for Building Programming”. While each of these projects project have explored some aspects of the contracted information exchanges needed to create an open standard for architectural programming, none of these projects have achieved a critical mass to be recognized as national standards and be widely implemented internationally.

The BPie specification is a direct result of the Building Programming IDM, of which the above mentioned projects have made their contribution to. Hence needs in their specified business processes should be covered by the BPie specification. This document will try to capture the differences in approach, scope and potential conflicts between the bindings to the BuildingSMART std. Ifc

Documents for The Building Programming IDM were published as draft documents the 5th of December 2011. The documents were submitted to buildingSMART International for a review and acceptance process to be finally distributed as a buildingSMART International IDM and now also as BPie specification. http://iug.buildingsmart.com/idms/information-delivery-manuals/idm-for-building-programming
# Table of Content

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>1</td>
</tr>
<tr>
<td>Conclusions and summary</td>
<td>3</td>
</tr>
<tr>
<td>IDM for Building Programming (2007 - 2011) - BPie</td>
<td>5</td>
</tr>
<tr>
<td>Portfolio and Asset Management : Performance Requirements (PAMPeR)</td>
<td>7</td>
</tr>
<tr>
<td>AR-5 project</td>
<td>8</td>
</tr>
<tr>
<td>PAMPeR, AR-5 and BPie</td>
<td>8</td>
</tr>
<tr>
<td>buildingSMART Aquarium on Room Data logistics</td>
<td>9</td>
</tr>
<tr>
<td>GSA Concept Design BIM 2010</td>
<td>10</td>
</tr>
<tr>
<td>Spatial Compliance Information Exchange (SCie)</td>
<td>13</td>
</tr>
<tr>
<td>Review comments</td>
<td>14</td>
</tr>
</tbody>
</table>
Conclusions and summary

This review and comments has revealed a need for some adjustments and supplements in the BPie specification. The IDM for building programming is not complete in the sense that not all the identified Exchange Requirements have been defined, e.g. ER for systems and equipment, and these exchange requirements are not covered by the BPie. First some clarifications;

The scope of the BPie specification is to cover some of the most vital and common requirements that are set to spaces and functions in the programming stage of a Building project, and maintain the requirements throughout design development, production to FM-handover. Requirements are not limited to the listed attributes. The intention is rather to agree of a list of commonly used requirements that can serve as an international standard, but the structure should be flexible for extensions.

ProcessMap: Due to different organizations and approaches to Building Programming (in different countries and by different clients) the ProcessMAP have been developed without a timeline to make it international robust. The format also indicates that programming is not necessarily something you do in a sequential development, but is part of an iterative process through design development – client requirement management.

Identification: One of the core concepts of the IFC standard is the GUID. All ifc objects shall have a global unique identifier that will never be reused. In a building project you would not need the space-/function ID to be globally unique, only project unique. Another relevant aspect is that the requirements space and the design space not necessarily is the same space object, but can be two different (linked) spaces and BIM authoring tools are not always able to import the GUID, read it to internal databases and re-export. The conclusion is that BPie uses a project unique ID (IfcSpace.name) as key ID. This ID comes in addition to a GUID, and its main purpose is to bind the “programmed space” together with the designed space in order to compare.

Basic/advanced: in the BPie documentation one had distinguished between a basic exchange and an advanced exchange (defined mandatory/optional for each). The basic exchange was defined as a minimum for a space list, while the advanced for defining a detailed RDS. The names of the columns have now been changed accordingly to “space list” and “RDS” as a more intuitive naming.

Mandatory/optional: The mandatory/optional columns for each exchange (space list/RDS) indicates whether this information is mandatory (M) or optional (O). Blank means that this field is not part of this exchange. Mandatory properties are the minimum for each exchange, but properties are mandatory as specified in projects if required in contract/by the client. One must distinguish between mandatory for certification of software (what should be technically possible to exchange) and mandatory for validation (what end users must add in a specific project/context). The Challenge 2013 is a validation where business rules are applied, and therefor requires specific data for some of the properties. E.g. it is mandatory to exchange occupancy information (Reference to the organization who will occupy the space) in the challenge 2013. In many (private) development projects, this information is not available at an early stage of the project and therefore cannot be exchanged. This property is then “mandatory as specified”.

Based on our review of previous efforts and received comments to the BPie specification, the following updates/adjustments will be made to the BPie specification:

1. There seem to be some confusion on terms and definitions, a glossary of terms have been added.
2. As a general rule, properties are mandatory as specified (in specific projects/by clients)
3. Change the definition for the project unique id for space (IfcSpace.name) to a more flexible definition. i.e. not required to express the main function (and department) relationship in the name/number.
4. Add “Custom” space properties. E.g. for client specific requirements or for special rooms etc.
5. Add new attributes with reference to document comments from “a US Government client” forwarded by Bill East.
   Some properties have for different reasons been left out of the specification. E.g. properties that relates to FF&E connections that will be part of another exchange (ER_Equipment)
6. The department/organizational aspects of zone have been added/moved as a separate entity/sheet, “Department_Function” to capture the characteristics of departments better. Zone remains, but now with the definition as a grouping of spaces that due to other shared characteristics are handled as “one”. E.g. the group share requirements or are part of a thermal zone, Security zone etc.
7. Proximity/location Requirements (new attributes) have been added to Space and departments,
   Based on comments from BPie Telecon and AR-5 review
   a. Set space proximity requirements (relations between functions)
      E.g.:
      i. Adjacent/Not adjacent
      ii. Same building/Not same building
      iii. Same story/-floor
      iv. Not same story/-floor
      v. Same wing/Not same wing
      vi. Vertically aligned (under/above)
      vii. Set maximum walking distance
      viii. Set minimum distance (point to point)
   b. Location (set required location for a function)
      i. Set building story requirement (e.g. a reception should be on ground floor)
      ii. Set required building location/orientation (e.g. function should face south-west)
8. Set required FF&E on space (FF&E list)
   ER_Exchange_requirement (equipment) are yet to be defined, but we might be able to define how the FF&E list should be expressed as space requirements.

Import: For import of BPie to design-/BIM authoring tools, one should define what’s relevant for different domains to import. E.g. an architect would need different properties than MEP consultants.
IDM for Building Programming (2007 - 2011) - BPie
This project is the latest project addressing the building programming/building brief process and relating design development. The project was initiated to address the building programming process and in particular client requirement management through design development. Previous process diagrams tend to describe programming as a sequential process, however in projects there’s a variety of approaches regarding level of detail and when things are defined. In “traditional” project development, programming is something you do prior to design, resulting in requirements that are not valid at project handover, and the “change-log” is held by the human resources involved. Especially in projects with a public client/building owner, the schedule from early stages to handover stretches over several years, staff changes and important project information get lost. Client requirement management is therefore a vital part of design development. This means that the programming process is redefined to become an iterative process where requirements and design evolves in parallel [see fig. 1]

![Fig. 1: The iterative process of Client requirement management](image)

To support this continuous development of requirements and make the IDM for Building Programming internationally robust and able to capture the needs of “all” projects, it was developed a ProcessMap (PM) without to many sequential dependencies [see fig.2.]
Basis and academic starting point for IDM for Programming

To capture the results and thoughts from the most relevant projects, participants from PAMPer and AR-5 project (Jeffrey Wix, Françoise Szigeti, Gerald Davis and Susan Nachtigall) was invited and participated in meetings/workshops.

The sponsoring partner of the project Statsbygg (the Norwegian GSA) was also a co-sponsor to the ConseqtDesignBIM 2010 project together with GSA and Senate Properties (Finnish GSA). The results from these two projects should therefore be aligned and equally compliant.

Further has Arto Kiviniemi been part of the project team, hence the result is also aligned with his Ph.D. dissertation: “Requirements Management Interface to Building Product Models“, GPA of courses 4.0 out of 4.0. CIFE, Civil and Environmental Engineering Department, Stanford University, USA, 2005.

The result from this project is the starting point of- and will be further developed as the BPie specification for “buildingSMART alliance January 2013 Challenge”,

Portfolio and Asset Management: Performance Requirements (PAMPeR)

The PAMPeR project has the focus of matching the demand- and supply-side, i.e. user/stakeholder requirements and programming/design solutions (or existing facilities).

Scope

The objective of this portion of the NBIMS is to create a standard view definition of these business functions and processes, using Industry Foundation Classes (IFC) for the exchange of information between demand-side requirements and the supply side views. The demand-side requirements are provided in performance terms by the users and other stakeholders. The supply side views are expressed in project planning, programming (e.g. the project brief), early design and subsequent down-stream applications. This demand-side information can be used for selecting from among proposed acquisition proposals, validating proposed design solutions, monitoring construction, auditing the “product” provided during commissioning and adjusting it during use. The owner’s and users requirements provide the benchmark to be achieved. This portion of the NBIMS precedes and complements the work of the ED project. See Figure 2 – Scope of PAMPeR and ED.

The PAMPeR project has many similarities with the IDM for Building Programming. The IDM has more detailed requirements, but PAMPeR requirements are covered by the BPie specification. This is commented more in detail below.
AR-5 project
The project description of the AR-5 project was as follows:

This proposed project will provide IFC capabilities to support early design processes starting from the interpretation of high level performance based requirements (to be provided by the PAMPeR project) through intermediate processes to hand-off to detailed design development. The intent is to be able to document requirements and early design decisions in a way that will make this information available in interoperable form to detailed design and subsequent processes in the project life cycle and to enable their tracking. Specifically, the project will:

- Identify and define the information objects that are used by owners and users to define what is required during the early design phase and to assess whether what is required has been provided in subsequent project phases. Performance requirements will be defined at a technical level, including: Size of space envelopes; special spaces; group proximities/separations; budget; technical requirements etc. Other project criteria will be documented.
- Develop IFC support for the following the four primary areas of consideration based on the requirements and criteria defined above:
  - Analysis of Alternatives
  - Blocking and Stacking
  - Bubble Diagramming
  - Review and Revision

The first bullet point above is reused from PAMPeR and directly overlapping IDM for Building Programming, while the second (and underlying points) are different/project specific.

PAMPeR, AR-5 and BPie
The results from PAMPeR, AR-5 and IDM for Building Programming are overlapping and complimentary. It is also obvious that they have been developed as continuations, where “the next” project uses the results from previous projects.

As a simplified summary, we think that the similarities and differences of the three projects are:

- PAMPeR defined the demand side (requirements) focusing on the facilities future production, feeding back the supply side (e.g. design solution/facility) in order to compare. In many ways similar to the IDM for Building Programming. However not as detailed to space requirements as the IDM project so we see no direct conflicts here.  
  End user, business driven approach
- AR-5 was also focused on the building programming, but also more in depth on the early stage design at an abstract level. This project also captured the interaction between early stage blocking and stacking and building programming, i.e. the proximity requirements between
rooms and functions. The concepts described in AR-5 is not in conflict with the BPie project, but actually supports many of the same ideas.

*Early stage design/Architect driven*

- IDM for programming deal with *client requirement management* controlling the project development and quality assurance processes. The project has flexible approach to how- and when requirements are developed, and have focused to capture a flexible “gross list” of requirements set to functions and spaces in a building project. It does not capture the proximity requirements from the AR-5 project but reuses the relevant Pset attributes defined in PAMPeR and AR-5.

*Client/Building owner, contractual, project management, facility lifecycle driven*

- All the projects have given contributions to development of the ifc schema.

To capture the proximity requirements identified in the AR-5 project, the BPie specification should be extended to cover also this business process/need.

**buildingSMART Aquarium on Room Data logistics**

This bS Aquarium took place during 2008 and 2009.

Regarding Process Maps (PM) and Exchange Requirements (ER) from the bS aquarium, there was created a draft for a PM and an ER for room data during the aquarium. These drafts were not finished or concluded during the aquarium and on the web they are marked

“DISCLAIMER:

This Exchange Requirement is a process artefact from the HOK Aquarium process

The status of the document is not an official standard document”


Our conclusion is that the PM, ER and MVD drafts that came out of the bS aquarium not are developed to a generic level that makes it natural to include in a detailed review or comparison to the proposed PM and ER for building programming. This conclusion is in agreement with the involved parties.
In developing the Concept Design BIM 2010, GSA partnered with Statsbygg (= GSA of Norway) and Senate properties (= the GSA of Finland), and the Construction Specifications Institute (CSI).

http://www.blis-project.org/IAI-MVD/MVDs/GSA-005/Overview.pdf

The scope was to optimize the design and analysis within four areas:

- Spatial Program Validation
- Human Circulation and security analysis
- Energy performance analysis
- Quantity takeoff and cost estimating

The relevant specification for BPIe is the Spatial Program Validation. As Statsbygg also was the main sponsoring partner in the IDM for Building Programming project, these two projects have been aligned. The main focus area of spatial program validation was design and analysis, and did not cover the programming stage/requirements BIM. There are therefore no conflicts to BPIe, but this project should enable GSA to validate that designers in GSA projects have met requirements set in the projects.

Looking at the PM for Spatial Program Validation, this is mainly reflected as the “ER Spatial Program Requirements” is the starting point of the process, and for dealing with requirement/design issues.

Fig. 3: PM - Spatial Program Validation
The key to the harmonization between GSA, Statsbygg and Senate was to agree to the scope of the ER’s (i.e. the data exchanges in- and between business processes). The process map above illustrates also some internal GSA processes/checkpoints, which is part of their Business rules, while Statsbygg and Senate have others.

It was also pointed out in the SCie project that the GSA Spatial Program Validation guide contains several agency-specific requirements that would not be appropriate for agencies outside the GSA. This is of course correct, so both the GSA Spatial Program Validation guide and Process Map are agency specific where the GSA Business rules are applied. Hence none of them serves as an international standard. Software vendors however would be more interested in the required data exchanges, the Exchange Requirements (ER’s), and these are made GSA independent and internationally robust.

One can conclude that GSA obviously are allowed to have their own GSA specific guidelines and processmap. The process map populated at http://www.blis-project.org/IAI-MVD/ (Architectural Design to Spatial Program Validation) should have been made GSA independent and more generic to serve as an international standard. The ER (ER_Spatial_Program_Validation) could however be valid in an international standard exchange.

Similar to the IDM for Building Programming project, CDB2010 has a very detailed specification for what to support for different entities. No conflicts have been identified between the definitions in the ER (ER_Spatial_Program_Validation) and the BPie definitions, but there are differences. The key for a successful exchange between CDB2010 and BPie would therefore be the space- and zone identification.

While BPie required (now changed) a project unique ID/Number that describes the main function, sub function/department and an enumeration for the space name (IfcSpace.Name), CDB2010 requires the space number in the same attribute. There is not necessarily a conflict between the two, but maybe the BPie definition should be made more flexible to cover other syntax’ for project unique ID’s.

Both projects uses IfcSpace.LongName for Space name.

![Diagram](image-url)  
*Fig. 4: CDB2010 attribute definitions for IfcSpace*
CDB2012 have detailed requirements for space classification that are covered in B Pie as support for multiple classifications to space. However not all the attributes are listed in the B Pie specification. It is important that one keeps this in mind when defining the binding to the Ifc schema. As an example space Type (Owner) http://www.blis-project.org/IAI-MVD/reporting/showConcept.php?CID=3386. If required in B Pie (mandatory), this must be defined as attributes. If this is business rules, this is covered by support for multiple classifications to space in B Pie.

Note: From the above one can see that GSA requires that Occupant organization is defined as a classification to IfcSpace. B Pie/COBi e requires that this is a classification to IfcZone.
Spatial Compliance Information Exchange (SCie)
The scope of SCie defined at http://buildingsmartalliance.org/index.php/projects/scie/

“...specification for a Spatial Compliance Information Exchange (SCIE) would be the COBIE specification with only those aspects of the COBIE specification that pertain to space lists, space function, space area measurements, and spatial zoning can be identified within a simplified COBIE spreadsheet. As a project progresses from design concept to construction completion and handover this data would need to be provided to ensure that the building was not deviating from the initial project scope.”

And further

“The 04-Space worksheet is the heart of the Spatial Compliance Information Exchange (SCIE). Within the Space worksheet the list of all rooms (and contiguous sub-spaces) are identified. Spaces are classified according to their OmniClass Room Function Codes. Application of data in the optional Attributes worksheet allows users to identify circulation, privacy, security, fire protection, historical preservation, and other zoning classification.”

Based on the above one can conclude; As the Building Program/Building brief “holds the initial project scope”, it is necessary to exchange BPie to fulfill the requirements in SCie. This also raises a question to point 5.g. in the document “COBie 2013 Challenge for Design and Construction Software-v04”:

5.g. Attribute Values of Spaces and Types shall carry through from BPie to COBie exchanges.

After the definitions in the BPie, requirements should be managed by requirement management tools, therefor one will have two parallel datasets, i.e. the requirements and the solution/design. It is important that the attribute values not necessarily have to be carried through by the BIM authoring tools.

To make BPie and SCie compliant, the required space/-zone attributes and classifications need to be compared/aligned. From the SCie scope: “Spaces are classified according to their OmniClass Room Function Codes”. To require OmniClass classification is a business rule for agencies/companies using OmniClass and should be changed to “Spaces are classified according to local classification for Room Function Codes, e.g. OmniClass”.
Review comments

The following review comments have been received:

Håkon Clausen, dRofus (original author of the Exchange Requirement)

**********************************************************

I do have some last comments and questions:

- **Building/Storey**: In my original ER documents there was in addition to the property listing, some text that explained for instance the minimal requirements for a valid exchange. As an implementer I feel that this is missing from this IDM (perhaps this should be moved to the PM document). For instance: Building has some mandatory fields in the excel sheet, but is building a mandatory object? In Norway, the building programmer does not create building/storeys, this is a design task and programming tools for instance has no information about these objects and will therefore never export these objects. There is no problem by having them in the ER if someone finds that useful, but is has to be optional.

- **Mandatory/Optional**: I do not quite understand how these columns should be interpreted. Again I feel the some text is missing that explains the excel sheet. I assume that a "blank" means that it's not part of the exchange, so when blank in basic but O/M in advances, it should not be part of a basic exchange. If this is true then all that are M/O in basic should also be part of advanced? If this is the case there are some errors in under the Space sheet, e.g. line 29, 31 and 37 (I have not checked the whole document)

- **Zone/Function**: It seems like "main function" has been renamed to "General zone requirements", leaving sub function reference main function that does not exists. Most of the property definition is also very general and are no good any more. I will suggest that we take some of the text from my original document back in as the text as it stand does not give any useful information to the implementers. We could simplify it by skipping the Main/Sub function and just call it "Function" since we have the Functional Decomposition property explains this. Also the Object Type = Function is important to be able to differentiate this zone among other zones. I suggest the following text:

  - Identification: Identification of the function for reference (M)
  - Name: The name of the function describing the intent/activity for this function (M)
  - Description: Description of the activity and performance of this main function (e.g. offices for 100 persons or x-ray department for 100 patients per day) (O)
  - Net area planned: Net area planned for this function. This could be given as a specific number or a percentage of the total. (O)
  - Classification: The function can be classified using a reference library or any national, standard or project specific classification. Multiple classifications can be used. (O)

- Zone Type: Function (M)

  - I will also suggest to move the Basic requirements to the top as it also contains things that are important to the functions.
  - ..and remove Space Decomposition from Zone (Basic requirements) as this has nothing to do with zones. Should be in the Space sheet (and it's already there).

Kimon Onuma, Onuma Planning System

**********************************************************
We think the overall structure and approach makes sense. We have a few comments. Sorry for jumping in late. Not sure if these topics have come up before.

On the document
IDM_exchange_req_building_programming_161111.pdf

Page 6

Space

General space requirements

Identification

We believe that space should be identified by a neutral identification and not tied to function and sub function.

We do agree that space should have a unique number for the project, but it should not be tied to the function.

Function and subfunction could be treated as an attribute or a zone rather than hard coding it into the space identification.

In the example given of:
A1.B1.001

If A1 is function and B1 is subfunction, what happens if the owner decides to change the function or subfunction of the space? Say from B1 to B2? If the space ID was more generic as 001 or 1001 and then the attributes or zones were used to identify the function and subfunction it would be easier to keep track of that unique space and change the assignment of function if needed. It would also allow for filtering and finding all Subfunctions B1 spaces, rather than having to parse it out of the space identification. Keeping this neutral would not limit specific owners or software vendors from using function and subfunction in the name if it is desired.

Page 20

"Zone" is not included in the Advanced column. Is this "Zone" similar in use as ifcZone or is it intended to be used INSTEAD of space in an early programming phase where space areas and attributes are still unknown? That would explain why it's not in the advanced column. It seems to us that it should be maintained in the Advanced column to be able to trace the function back to this zone.

Lawrence Ciscon, “Larry”, Trelligence

*********************************

Overall what you sent looks good to us. We’re more concerned with what appears to be missing. Below is a list of low and high priority omissions that we can see. It may be that these are covered in other parts of the specification. In that case it still might be good to include at least a reference to them in this document to present a complete process model. We are happy to provide additional thoughts/input/ideas on any or all of these if it's useful.

One general comment: we share Kimon's concern about the specification of the Identification field of the Space. However my guess is that implicit in this definition is also the concept of a Global ID for each Space (and every object, for that matter) separate from this Identification field. The Global ID
would ensure traceability on edits and changes, especially to the Identification field itself. Even if the Global ID is implicitly included in all of these definitions thru inheritance, it would be good to include it on these sheets for clarity and completeness.

High Priority

-----------------

1. Custom Space properties

Although your list of Space properties is very extensive, it does not cover all conditions by any means – especially for specialty room types. It is extremely important to include some ability to include project-specific properties for the Spaces. This could be very simple to begin with (e.g. simple text fields). "Leave a spot to put what you forgot"

2. Room contents (e.g. Furniture & equipment)

Maybe I'm missing something obvious, but I don't see any Equipment/Furniture models, or any way of attaching those models to the Spaces?? This seems like a major shortcoming. At least 90% of the projects we're involved with involve both Spatial requirements and some Furniture/Equipment requirements from the very earliest stages of the programming effort.

Note: I realize the fields for quantities of windows, doors, etc. in the Spaces fill some of this need and provide the most efficient storage model. But in our experience the kinds of rooms where you care about specific contents tend to be non-normal and thus cannot be included in a standardized list of contents. Thus you need both approaches.

Med/Low Priority

-----------------

1. Departments vs. Zones

It is important to track Departments as distinct from Zones. I realize that you can express a Department as a Zone, but there are some characteristics of Departments that are not shared by Zones, e.g. Grossing factors, hierarchical Department relationships, inter-Department relationships, etc. It may be a matter of simply deriving Departments from Zones and adding some additional properties.

2. Additional programmatic constraint types

It would be nice at some point to add some additional constraint specifications, e.g. Rooms based upon capacity rather than a static size; Service areas as a % of the overall program area; related constraints (X rooms for every Y rooms), etc.

Kimon Onuma, Onuma Planning System

**************************************************
Yes, I assume the GUID as defined in IFC specs still holds true here. The GUID would be random and unique to each component or space and therefore should not be used as a way to track room numbers.

We think room numbers should just be simple IDs that can be understood by owners. They can be in the format that shows the subfunction, but that should not be a requirement.

So our recommendation is:
- GUIDs are part of the space by default, as defined in IFCs
- Room numbers can be any unique number / letter but they should not be required to manage other attributes such as subfunction.

I agree on your comments for the high priority items.
We do not see a spot in the IDM for custom space properties. We are not sure if the function of an IDM is to describe all possible entities that you would be allowed to include. If that’s the case, there should definitely be an open section for these.

### Participant BPie Telecon – discussion (Informational Meeting for Design Software)

Question/comment about missing proximity requirements.

**Dr. Luciana Burdi, Manager, Lean and Integrated Building Solutions (BIM/IPD)**

PROCESS MAP:
The process map per-se is very useful, but it is very convoluted to read, follow and understand it. When you try to explain “who does what” the message is not very clear. Something simpler to read and follow a timeline might benefit the map.

IDM FOR BUILDING PROGRAMMING:
This is very close to my heart as I have been working on a programming office for 7 years. So, I am very happy to finally see something that relates BIM and programming

- **Page 1:** Please clarify what “classification” stands for. A glossary of terms might be beneficial because it appears to me that noun are used different times with different meaning.
- **Page 3:** In the Building general requirements, you should add “building Number” having only the name might not be enough
  - “Classification” needs to be better defined. Here we are with the same term used in a different context.
  - Under “Occupancy requirements” we should ask to identify who the “owner” is and who “manages” the building – they can be two separate entities
- **Page 5:** Story = Floor (?) again need for a glossary of terms
- **Page 6:** Space= Room
  - Under General Space requirements we should add: Room number (which can be different from room name)
  - Add also who “uses the space”, who “owns the space” and who “maintain the space” – they can be all separate entities
- **Page 7**: External or Internal Space: it is not clear to me what this mean
- **Page 8**: within the Occupancy requirements you should include who “uses the space”, who “owns the space” and who “maintain the space” – they can be all separate entities

And as a general comment, I think it is missing a whole section on **Accessibility** requirements

---

**Comments from “from a US Government client” forwarded by Bill East**

Comments added in separate document. Will be handled directly in the ER update. Comments related to: room ID, questions to definitions and examples, terms and definitions (Functiones, departments, Occupancy, tenants, etc.), FF&E (status).

Suggested also some new and relevant attributes, and most of them will be added to the final ER.

---

**Bill East (Telecon with Rolf)**

Each space must have information about occupancy (who will occupy the space after handover) and a space classification (Mandatory).