Energy analysis and client requirements in early design

Nils Outters, NCC Construction, Sweden

bips conference, Nyborg Dk

2010-09-07

Client needs regarding energy

An attractive building should be:

- Cost effective
- Resource efficient
- Comfortable for end users

How do we handle these needs?

A design team should use our integrated Inpro life-cycle design supported by our model server solution

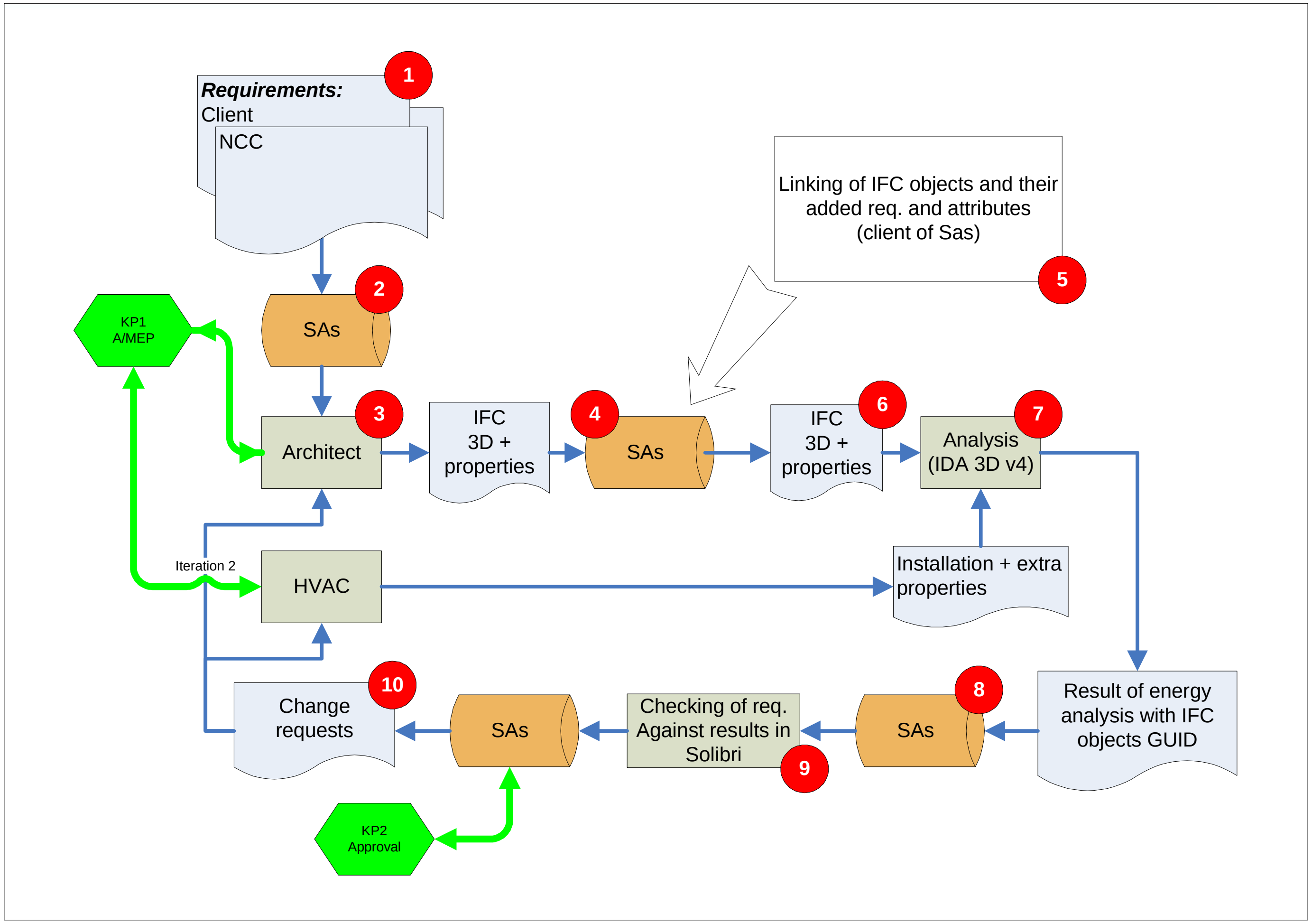
- Focus on early design
- Management of life-cycle
- BIM oriented
- IFC-based
- Multidisciplinary cooperation
- Tested on small scale real life scenario

(Courtesy of J-CDS: www.j-cds.nl)

An intuitive solution

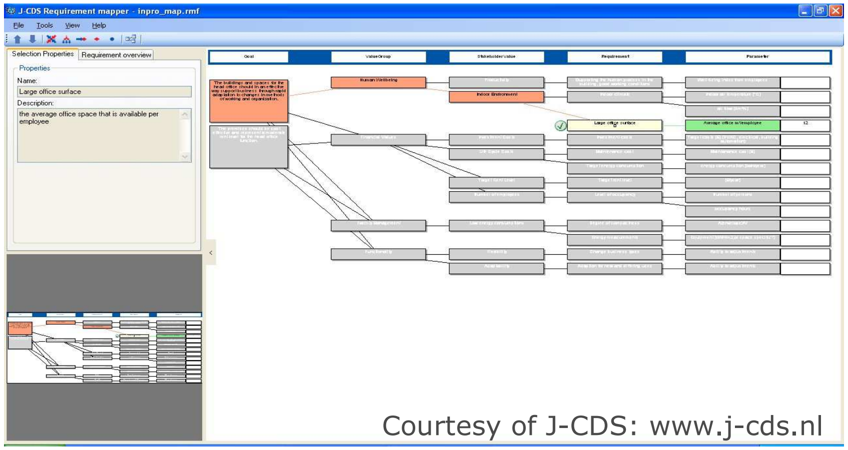
1. Capture client energy and climate requirements *imported* and *mapped* into the model server
2. Architect and HVAC-design team get necessary requirements from the model server
3. Architect and HVAC design *imported*, *approved, consolidated* and *mapped* in the model server through IFC
4. Energy specialist exports in IFC a model suitable for energy analyses
5. Energy analyses results *imported*, *consolidated* and *mapped* in the model server through IFC
6. Validation of the energy analysis results against client requirements.
7. *Approval of design or design iteration*

Data flow chart



Mapping of new requirements relatively easy

- InPro framework to capture stakeholder values was developed in briefing process: starting from goals of the client, capture stakeholder values, requirements and parameters
- Concurrent Design (CD) as enabling method
- In CD sessions using the J-CDS CDP or the Requirements Mapper
  - Involve all stakeholders from the beginning of the project
  - Exchange information between the team members
- Allows for systematic handling of requirements



Courtesy of J-CDS: www.j-cds.nl

"Non digitally structured" requirements

- Document based requirements is challenging. How do we do?

NCC

NCC Property Development

PROGRAMKRAV – SVENSKT KONTORSHUS

- Sekvensstyrning värme-kyla lämnas som optionspris.
- Energiklass A för installerade motorer.

Energieffektivitet åstadkommes genom att projektet uppfyller GreenBuilding – kraven.

B.2.4 Inomhusklimat

- Innetemperatur: Svenska inneklimatinstitutets kravnivå TQ2.
  Dimensionerande utetemperatur skall vara :
  Sommar: +27 °C vid RH 50 %
  Vinter: -2 °C under DUT (för Stockholm blir det -20 °C)
  Vid ev. varmgarage skall temperaturen vara +14 °C
- Komfortkylsystem skall dimensioneras för en internlast av 30 W/m² LOA
- Hygienluftlöde: min 1,5 l/s,m² för kontor/kontorslandskap och 4,5 l/s,m² för konferens-/mötesrum. Återluft får ej förekomma till kontor och butiker. Återluft från kontor får användas för att ventilera garage.
- Frånluftsdon bör placeras vid kopiering/skrivare.
- Koldioxid 1 h medelvärde: max 1000 ppm (800 ppm skall dock eftersträvas vid uteluft på 400ppm)
- Radon: max 200 Bq/kbm, årsmedelvärde
- Formaldehyd: <0,05 mg/m³
- Kolmonoxid: <0,5 mg/m³
- Filterklass lägst F7 i både till- och frånluftskanaler
- Miljöhänsyn tas genom tillämpning av NCCs miljöregler och tillämpning av BASTA-kriterier.
- Deklarerad egenemission ska vara lägre än 30 µg/m²,h från färg, avjämningsmassor, golv- och vägglim. Från golvmaterial ska egenemissionen vara lägre än 100 µg/m²,h
- Heltäckningsmattor skall vara testade och godkända m a p astma och allergier
- Elektromagnetiska fält i fastighetens grundinstallation: <0.2 µT (5Hz-2kHz), <10 V/m RMS (5Hz-2kHz). Gäller för stadigvarande arbetsplats. (Resten lämnas som optionpris)
- Elanläggning skall utföras som 5-ledarsystem och vara spänningsutjämnat.
- AQ2 och NQ2 gäller som lägsta krav för ljud respektive emissioner.

B.3 Säkerhet och Trygghet

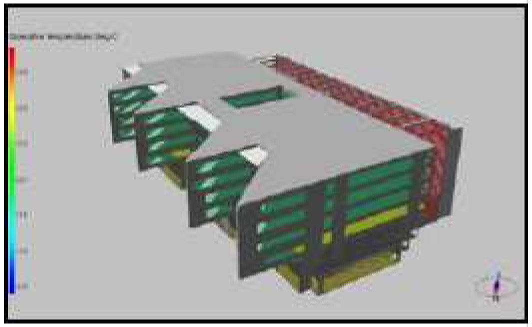
Den övergripande policyn lyder:

> *"Väl genomtänkta säkerhetslösningar skapar trygghet för individer och företag. Hög säkerhetsnivå i byggnader samt en verksamhet som generar rörelse även efter normal kontorstid bidrar till en trygg miljö"*

Dokumentnamn	Datum för upprättande	Senast ändrad	7 (43)
Programkrav_Kontorshus	2006-12-15	2008-01-31	

IDA Indoor Climate and Energy 4.0

- Dynamic multi-zones simulation application
- Accurate study of thermal indoor climate of individual zones
- Energy consumption of the entire building



Energy analysis in IDA v4

- Import of zone geometry by IFC into IDA
- Energy use scenario based on data from the model server
- Conversion of IDA results to IFC property sets with a *Python* script

```
pset;;Pset_buildingEnergy;Building;2hQBAVPOr5VxhS3Jl0O47h;FZK-Haus
value;Integrated energy;183691.6;

pset;;Pset_SpaceThermalCalculated;Space;0e_hbkIQ5DMQlIJ$2V3j_m;bad
value;SpaceTemperatureMax;25.736;DEGREE_CELSIUS
pset;;Pset_SpaceThermalCalculated;Space;2RSCzLOBz4FAK$_wE8VckM;buero
value;SpaceTemperatureMax;26.642;DEGREE_CELSIUS
pset;;Pset_SpaceThermalCalculated;Space;3$f2p7VyLB7eox67SA_zKE;flur
value;SpaceTemperatureMax;25.453;DEGREE_CELSIUS
pset;;Pset_SpaceThermalCalculated;Space;2dQFggKBb1fOc1CqZDIDlx;galerie
value;SpaceTemperatureMax;25.649;DEGREE_CELSIUS
pset;;Pset_SpaceThermalCalculated;Space;17JZcMFrf5tOftUTidA0d3;k_s__che
value;SpaceTemperatureMax;26.608;DEGREE_CELSIUS
pset;;Pset_SpaceThermalCalculated;Space;347jFE2yX7IhCEIALmupEH;schlafzimmer
value;SpaceTemperatureMax;26.219;DEGREE_CELSIUS
pset;;Pset_SpaceThermalCalculated;Space;0Lt8gR_E9ESeGH5uY_g9e9;wohnen
value;SpaceTemperatureMax;26.327;DEGREE_CELSIUS
```

```python
def GetIFCObjectGUIDfromIFC(strIFC,strRoomName):
    with open(strIFC, 'r') as f:
        for line in f:
            IFCSPACEIndex = string.find(line,'IFCSPACE')
            intLineLength = len(line)
            if (IFCSPACEIndex>-1) & (string.find(line.lower(),strRoomName.lower())>-1):
                words = line[IFCSPACEIndex+9:intLineLength-3].split(',')
                return words[0]
        f.close()
    return

def GetBuildingGUIDfromIFC(strIFC):
    with open(strIFC, 'r') as f:
        for line in f:
            IFCSPACEIndex = string.find(line,'IFCBUILDING')
            intLineLength = len(line)
            if (IFCSPACEIndex>-1):
                words = line[IFCSPACEIndex+12:intLineLength-3].split(',')
                return [words[0],words[2]]
        f.close()
    return

def GetIntegratedEnergy(strSuppliedEnergy):
    integratedEnergy = 0
    with open(strSuppliedEnergy, 'r') as f:
        f.readline()
        for line in f:
            words = line.split()
            for energy in words[2:]:
                integratedEnergy += float(energy)
        f.close()
    return integratedEnergy
```

Conclusions

- Energy analysis supported by a model server has been demonstrated
- Requirements, design and analyses are handled in an objective and systematic manner
- Tools and methods allow for traceable design iterations
- Integration with other key processes of the early design has been established within Inpro