

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ  
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη ευφυούς συστήματος για την εφαρμογή πολιτικών σε έξυπνα  
ενεργειακά κτίρια με χρήση υπολογιστικής επιχειρηματολογίας**



ΠΑΝΤΕΛΕΗΜΩΝ ΚΡΙΝΑΚΗΣ

ΕΠΙΒΛΕΠΩΝ:

Καθηγητής Νικόλαος Ματσατσίνης, σχολή ΜΠΔ

ΜΕΛΗ ΕΞΕΤΑΤΙΚΗΣ ΕΠΙΤΡΟΠΗΣ:

Μέλος ΕΔΙΠ Δρ. Νικόλαος Σπανουδάκης, σχολή ΜΠΔ

Καθηγήτρια Διονυσία Κολοκοτσά, σχολή ΧΗΜΗΠΕΡ

**Χανιά, 2021**



TECHNICAL UNIVERSITY OF CRETE  
SCHOOL OF PRODUCTION ENGINEERING AND MANAGEMENT

THESIS

**Development of an intelligent system for decision policy into smart  
buildings with the use of hierarchical argumentation theory**



**PANTELEIMON KRINAKIS**

SUPERVISOR

Professor Nikolaos Matsatisnis, School of PEM

THESIS COMMITTEE:

Researcher-Teaching assistant Dr. Nikolaos Spanoudakis, School of PEM

Professor Dionysia Kolokotsa, School of CHENVENG

**Chania, 2021**

## Ευχαριστίες

Η εμπιστοσύνη που μου έδειξε το μέλος ΕΔΠ Δρ. Νικόλαος Σπανουδάκης για την ανάθεση ενός τέτοιου έργου, που ξέφευγε από τα όρια της σχολής μου, ήταν πραγματικά μεγάλης σημασίας για εμένα. Οφείλω λοιπόν σε αυτόν, ένα μεγάλο ευχαριστώ από καρδιάς για τις ατελείωτες ώρες συνεργασίας και καθοδήγησης που μου αφιέρωσε. Μιας καθοδήγησης που ξεπέρασε κατά πολύ τα όρια της διπλωματικής και έφτασε μέχρι ζητήματα καριέρας αλλά και προσωπικότητας. Όλος ο χρόνος που πέρασα κοντά του στο εργαστήριο ήταν ένα μεγάλο μάθημα για εμένα.

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον καθηγητή κ. Νικόλαο Ματσατσίνη που είχε την υψηλή εποπτεία της εργασίας. Επίσης να ευχαριστήσω την καθηγήτρια της σχολής ΧΗΜΗΠΕΡ, κα. Διονυσία Κολοκοτσά για την καθοδήγηση στα ενεργειακά ζητήματα και τις προτάσεις της που διαμόρφωσαν και έδωσαν μια μεγαλύτερη διεπιστημονικότητα στην εργασία αυτή.

Δεν θα μπορούσα φυσικά να μην ευχαριστήσω από καρδιάς όλους μου τους φίλους για τις υπέροχες στιγμές που ζήσαμε στα Χανιά. Ιδίως κατά την διάρκεια της διπλωματικής, η στήριξη που μου έδειξαν ήταν ουσιαστική, αφουγκραζόμενοι τα προβλήματα και τις σκέψεις μου και βοηθώντας με να προχωρήσω.

Τέλος θα ήθελα να ευχαριστήσω θερμά την οικογένεια μου και κυρίως του γονείς μου που έμπρακτα εδώ και έξι χρόνια ήταν δίπλα μου σε κάθε στιγμή, καλή ή κακή. Να με συμβουλεύουν και να με κάνουν να πιστεύω στον εαυτό μου στις δύσκολες στιγμές. Η απέραντη αγάπη και εμπιστοσύνη που μου δείχνουν είναι πάντα για να μπορέσω να συνεχίζω υπερπηδώντας κάθε εμπόδιο στον δρόμο μου.

Το Πολυτεχνείο Κρήτης υπήρξε για εμένα μια μεγάλη οικογένεια που με καλωσόρισε από την πρώτη κιόλας μέρα με τον πιο θερμό τρόπο. Νιώθω ιδιαίτερη ευγνωμοσύνη απέναντι σε όλους τους ανθρώπους με τους οποίους συνεργάστηκα αυτά τα χρόνια μέσω του πολυτεχνείου. Όλες αυτές οι εμπειρίες θα μείνουν για πάντα χαραγμένες στην μνήμη μου και ακόμη περισσότερο, αφού διαμόρφωσαν σημαντικά τον άνθρωπο που είμαι σήμερα.

Κάθε τέλος και μια καινούρια αρχή...



---

## ΠΕΡΙΛΗΨΗ

---

Αντικείμενο της παρούσας εργασίας είναι ο καθορισμός ενεργειακής πολιτικής σε έξυπνα κτίρια. Στόχος είναι η λήψη αποφάσεων σε ποικίλα ζητήματα στα οποία υπάρχει σύγκρουση πολιτικών μεταξύ των προτεραιοτήτων διαφορετικών χρηστών και ρόλων. Μέσω αυτών των πολιτικών αυτοματοποιείται η διαδικασία λήψης αποφάσεων και ταυτόχρονα επιτυγχάνεται αυξημένη ενεργειακή αλλά και οικονομική αποδοτικότητα. Για την υλοποίηση της εργασίας γίνεται χρήση αισθητήρων IoT (Internet of Thing) για την λήψη αναγκαίων μετρήσεων και δεδομένων των οποίων η διασύνδεση γίνεται μέσω του προτύπου KNX. Επιπλέον η μοντελοποίηση του κτιρίου γίνεται μέσω της εφαρμογής Open Studio/Energy plus. Για την επεξεργασία των δεδομένων και την λήψη απόφασης χρησιμοποιείται ένα εργαλείο ιεραρχικής επιχειρηματολογίας, ο Gorgias. Η μελέτη περίπτωσης γίνεται για το κτίριο των Μηχανικών Περιβάλλοντος του Πολυτεχνείου Κρήτης όπου υπάρχουν τοποθετημένοι κατάλληλοι αισθητήρες για την μέτρηση βασικών παραμέτρων-δεδομένων, όπως θερμοκρασίας, φωτισμού και επιπέδων διοξειδίου του άνθρακα. Για την μελέτη περίπτωσης χρησιμοποιούνται ως παραδείγματα διάφορες καταστάσεις, οι οποίες δεν ήταν δυνατόν να επιλυθούν αποτελεσματικά με την χρήση υφιστάμενων εργαλείων και προγραμμάτων. Το αποτέλεσμα της εργασίας είναι η διαμόρφωση ενεργειακών πολιτικών που βοηθούν στην μείωση της κατανάλωσης ενέργειας καθώς και η αποτελεσματική διαχείριση καταστάσεων όπου υπάρχει σύγκρουση αποφάσεων.

---

# ABSTRACT

---

The objective is to define and apply energy policies into smart buildings. The goal is to make decisions on a variety of issues where there is a conflict of policies between the priorities of different users and roles. Through these policies, the decision-making process is automated and at the same time increased energy and economic efficiency is achieved. For the implementation of the work, IoT (Internet of Things) sensors are used to take the necessary measurements and data which are interconnected through the KNX standard. The relevant building's energy consumption data is imported by Open Studio/Energy plus application. A tool for hierarchical argumentation, Gorgias, is used to process the data and make decisions. The case study is applied on the building of Environmental Engineers of the Technical University of Crete where there are suitable sensors for measuring basic parameters-data, such as temperature, lightings and CO<sub>2</sub> levels. Various situations are used as case studies, which could not be solved efficiently by using existing tools and software. The result of the work is the implementation of energy policies that help to reduce energy consumption as well as the effective management of situations where there is a conflict of decisions.

## Περιεχόμενα

Κατάλογος εικόνων .....	4
Κατάλογος πινάκων.....	5
Συντομογραφίες.....	5
Κεφάλαιο 1° Εισαγωγή.....	6
1.1 Υφιστάμενη κατάσταση .....	6
1.2 Κενά που προκύπτουν-Προβλήματα .....	7
Κεφάλαιο 2° Θεωρητικό υπόβαθρο.....	8
2.1 Gorgias Framework .....	8
2.1.1 Κανόνες του Gorgia.....	8
2.1.2 Gorgias Cloud.....	9
2.2 Πρωτόκολλο KNX .....	10
2.3 Διεπαφή Προγραμματισμού Εφαρμογών (API) .....	11
2.4 Representational State Transfer (REST) .....	12
2.5 Swagger .....	12
2.6 PyCharm.....	13
2.7 Django .....	13
2.8 Γραφικό Περιβάλλον Χρήστη- Graphic User Interface .....	15
2.9 HTML forms .....	16
2.10 Engineering Tool Software 5.....	17
2.11 Sketch Up- Open studio.....	18
Κεφάλαιο 3° Προβληματική-Στόχοι εργασίας .....	20
3.1 Ο Γοργίας .....	20
3.2 Γενικοί στόχοι .....	20
3.3 Περιθώρια εξέλιξης .....	21
3.4 Δικλίδες ασφαλείας.....	21
Κεφάλαιο 4° Ανάπτυξη εφαρμογής.....	22
4.1 Αρχιτεκτονική .....	22
4.2 REST API Gorgias Cloud .....	23
4.3 REST API BAOS 777 .....	25
4.4 Εφαρμογή του Django.....	28
4.5 Το γραφικό περιβάλλον του συστήματος.....	29
4.6 Παραμετροποίηση επιλογών στο Open Studio .....	36



4.6.1	Αποτελέσματα από το Open Studio .....	38
4.7	Ανάπτυξη κανόνων στον Gorgia .....	41
4.7.1	User's policy .....	42
4.7.2	Energy management policy .....	42
4.7.3	Emergency policy .....	43
4.7.4	Combined data .....	44
4.7.5	Meta data policy .....	44
4.7.6	Background file .....	45
Κεφάλαιο 5°	Πιλοτική εφαρμογή .....	49
5.1	Το Πολυτεχνείο Κρήτης .....	49
5.2	Κτίριο K2 της Σχολής Μηχανικών Περιβάλλοντος .....	49
5.3	Χρήση με πραγματικά δεδομένα .....	51
5.4	Δεδομένα από το KNX BAOS .....	52
5.5	Εφαρμογή σεναρίων .....	54
	Σενάριο 1° .....	55
	Σενάριο 2° .....	57
	Σενάριο 3° .....	58
	Σενάριο 4° .....	59
	Σενάριο 5° .....	60
	Σενάριο 6° .....	61
	Σενάριο 7° .....	63
	Σενάριο 8° .....	64
	Σενάριο 9° .....	65
	Σενάριο 10° .....	66
Κεφάλαιο 6°	Αποτελέσματα εργασίας .....	68
Κεφάλαιο 7°	Σύνοψη και μελλοντική εργασία .....	70
Βιβλιογραφία .....		72
Appendix .....		0
KNX BAOS REST API .....		0
Gorgias REST API .....		1
HTLM Form .....		6
Gorgias files .....		10
Background policy .....		10

User Policy .....	12
Energy Management Policy .....	13
Emergency Policy.....	14
Meta-Data Policy.....	14
Combined Policy .....	15

## Κατάλογος εικόνων

EIKONA 1. Σύνολο Αρχείων στο Gorgias cloud.....	9
EIKONA 2. Το Execution panel του Gorgia.....	10
EIKONA 3. Το γραφικό περιβάλλον των windows 10.....	16
EIKONA 4. Η βασική οθόνη επιλογών του ETS5.....	18
EIKONA 5. Το κτήριο K2 μοντελοποιημένο στο Sketch Up .....	19
EIKONA 6. Η οθόνη επιλογών του Open Studio.....	19
EIKONA 7. Διάγραμμα Αρχιτεκτονικής της εφαρμογής.....	22
EIKONA 8. Ρυθμίσεις εκτέλεσης του Django .....	29
EIKONA 9. Βασικό Μενού επιλογών του εργαλείου .....	33
EIKONA 10. Οι διαφορετικές θερμικές ζώνες του κτιρίου .....	38
EIKONA 11. Αεροφωτογραφία της πολυτεχνειούπολης .....	49
EIKONA 12. Το κτίριο K2 της Σχολής Μηχανικών Περιβάλλοντος .....	50
EIKONA 13. Κεντρικό μενού επιλογών του BAOS 777 .....	53
EIKONA 14. Ρυθμίσεις Δικτύου BAOS 777 .....	53
EIKONA 15. Συνδεδεμένες συσκευές στο BAOS 777 .....	54
EIKONA 16. Σενάριο 1.....	56
EIKONA 17. Σενάριο 2.....	57
EIKONA 18. Σενάριο 3.....	58
EIKONA 19. Σενάριο 4.....	60
EIKONA 20. Σενάριο 5.....	61
EIKONA 21. Σενάριο 6.....	62
EIKONA 22. Σενάριο 7.....	64
EIKONA 23. Σενάριο 8.....	65
EIKONA 24. Σενάριο 9.....	66
EIKONA 25. Σενάριο 10.....	67

## Κατάλογος πινάκων

ΠΙΝΑΚΑΣ 1. Συνολικές καταναλώσεις ανά είδος.....	39
ΠΙΝΑΚΑΣ 2. Καταναλώσεις ανά θερμική ζώνη .....	40
ΠΙΝΑΚΑΣ 3. Κατάταξη θερμικών ζωνών .....	41

## Συντομογραφίες

**OSI:** Open Systems Interconnection

**HTML:** HyperText Markup Language

**XML:** Extensible Markup Language

**JSON:** JavaScript Object Notation

**HTTP:** Hypertext Transfer Protocol Secure

**RPC:** Remote Procedure Calls

**SOAP:** Simple Object Access Protocol

**WSDL:** Web Services Description Language

**WAF:** Web Application Framework

**BSD:** Berkeley Software Distribution

**SQL:** Structured Query Language

**URL:** Uniform Resource Locator

**IDE:** Integrated development environment

**VPM:** volume per million

**UI:** User Interface

**IoT:** Internet of Things

**ETS:** Engineering Tool Software

**GIS:** Geographic Information Systems

# Κεφάλαιο 1<sup>ο</sup>

## Εισαγωγή

Οι τεχνολογίες στο χώρο των έξυπνων κτιρίων αλλά και στην λήψη αποφάσεων βρίσκονται σε μια συνεχή ανάπτυξη με σκοπό την βελτιστοποίηση του συνόλου των στοιχείων που τις αποτελούν. Κύριος σκοπός αυτής την ανάπτυξης είναι η περαιτέρω εξοικονόμηση ενέργειας στα κτίρια, τόσο για ιδιωτικές κατοικίες όσο και μεγάλα κτίρια, όπως νοσοκομεία, εργοστάσια και πανεπιστήμια.

Με την πάροδο των ετών και την ραγδαία τεχνολογική ανάπτυξη, ο τομέας τόσο των “έξυπνων” αισθητήρων όσο και της μετάδοσης σημάτων βρίσκονται στο επίκεντρο. Με την χρήση τεχνολογιών όπως του δικτύου 4G και την εξέλιξη αυτού, το 5G που ήδη σιγά σιγά μπαίνει στην καθημερινότητα μας, οι ταχύτητες αποστολής αλλά και οι λήψεις σημάτων έχουν αυξηθεί σημαντικά. Ταυτόχρονα, στον τομέα του hardware γίνεται ανάπτυξη πιο προηγμένων αισθητήρων με νέες δυνατότητες, μικρότερο μέγεθος αλλά και χαμηλότερο κόστος. Αυτές οι τεχνολογίες συνεπώς οδηγούν και στην ανάπτυξη του δικτύου IoT [1] (Internet of Things). Το IoT είναι ένα δίκτυο συσκευών που έχουν ως βασικό χαρακτηριστικό την ικανότητα πρόσβασης στο διαδίκτυο. Μέσω αυτής της τεχνολογίας οι συσκευές μπορούν να αλληλεπιδρούν μεταξύ τους με σκοπό την ανταλλαγή πληροφοριών και δεδομένων και απώτερο στόχο, τον έλεγχο αυτών από τον χρήστη. Υπάρχει μια μεγάλη ποικιλία διαφορετικών συσκευών που μπορούν να διαθέτουν το πρότυπο IoT, όπως τα οχήματα με αισθητήρες, κάμερες ασφαλείας και αισθητήρες για έξυπνα σπίτια. Με αυτό τον τρόπο, πολλά διαφορετικά δεδομένα από διαφορετικές κατηγορίες συσκευών συλλέγονται και επεξεργάζονται κατάλληλα ώστε να αντιμετωπιστεί μια συγκεκριμένη ανάγκη.

### 1.1 Υφιστάμενη κατάσταση

Ο τομέας των έξυπνων κτιρίων παρουσιάζει μεγάλη ανάπτυξη τα τελευταία χρόνια τόσο αναφορικά με την χρήση νέων αισθητήρων όσο και την αποτελεσματικότερη λήψη αποφάσεων. Αυτή την στιγμή ένα από τα αποτελεσματικότερα εργαλεία που χρησιμοποιείται στα έξυπνα κτίρια είναι το λογισμικό ETS5 (Engineering Tool Software 5). Αυτό το λογισμικό χρησιμοποιείται ευρέως για την διασύνδεση συσκευών που βασίζονται στο πρότυπο KNX. Το KNX, είναι ένα ανοιχτό πρότυπο επικοινωνίας συσκευών και αισθητήρων. Οι πλέον εξελιγμένες εταιρείες στον χώρο κατασκευάζουν συσκευές που συμβαδίζουν με το πρότυπο KNX ώστε να υπάρχει καθολική εφαρμογή και υποστήριξη των διαφορετικών συσκευών.

Επιπλέον για την διαχείριση της ενέργειας σε κτιριακά συγκροτήματα, όπως Πολυτεχνειούπολεις, έχουν αναπτυχθεί αντίστοιχα συστήματα. Ένα από αυτά είναι η πλατφόρμα “CAMP-IT” [2], όπου, μέσω ποικίλων αισθητήρων, λαμβάνονται δεδομένα και μέσω αλγορίθμων τεχνητής νοημοσύνης ασαφούς λογικής (fuzzy login) λαμβάνονται αποφάσεις. Στόχος αυτού του συστήματος είναι η αποτελεσματική μείωση κατανάλωσης ενέργειας των κτιρίων.

Μια ακόμη εφαρμογή αποτελεί η πλατφόρμα Γεωγραφικού Συστήματος Πληροφοριών (GIS) για την παρακολούθηση κτιρίων μηδενικής ενεργειακής κατανάλωσης [3]. Μέσω αυτής της πλατφόρμας γίνεται ενοποίηση και παρακολούθηση δεδομένων από αισθητήρες που βρίσκονται σε τέσσερις διαφορετικές χώρες. Αξίζει να σημειωθεί ότι οι αισθητήρες που χρησιμοποιούνται σε αυτό το σύστημα είναι τύπου KNX, αντίστοιχοι με αυτούς που χρησιμοποιήθηκαν στην παρούσα εργασία.

Τέλος, έχει αναπτυχθεί ένα σύστημα για την λήψη αποφάσεων σε έξυπνα κτίρια, βασισμένο σε defeasible logic [4]. Μέσω αυτού του συστήματος γίνεται διασύνδεση ενός συνόλου αισθητήρων που βρίσκονται σε ένα κτίριο και μέσω defeasible logic γίνεται λήψη αποφάσεων. Με αυτό τον τρόπο επιτυγχάνεται μια ενεργειακή εξοικονόμηση 4% σε καθημερινή βάση.

## 1.2 Κενά που προκύπτουν-Προβλήματα

Οι τεχνολογίες που εφαρμόζονται σήμερα διαθέτουν μεγάλα περιθώρια βελτίωσης. Το πρόγραμμα ETS5, που είναι ένα από τα πλέον διαδεδομένα προγράμματα για την διασύνδεση αισθητήρων και την λήψη αποφάσεων από αυτούς, διαθέτει ένα περίπλοκο περιβάλλον. Συνεπώς, για να μπορέσει κάποιος να χειριστεί το πρόγραμμα, απαιτούνται γνώσεις και συνήθως χρησιμοποιούνται από εξειδικευμένα άτομα. Ταυτόχρονα, καθώς οι επιλογές και οι ρυθμίσεις είναι περιορισμένες, μπορούν να ληφθούν μόνο βασικές αποφάσεις με λίγες παραμέτρους και χωρίς το συνδυασμό που μπορεί να απαιτεί μια ολοκληρωμένη εφαρμογή. Για την δημιουργία λοιπόν ενός πολυεπίπεδου σεναρίου χρησιμοποιώντας εξαντλητική απαρίθμηση όλων των πιθανών επιλογών (If...then...else), απαιτείται ένα ιδιαίτερα μακροσκελές πρόγραμμα. Το πρόγραμμα αυτό χρειάζεται ικανό χρόνο για να υλοποιηθεί και ξεκάθαρη αναγνώριση και χαρτογράφηση των αναγκών εξαρχής. Για αυτό τον λόγο η ενσωμάτωση νέων πολιτικών και κανόνων είναι αδύνατη χωρίς της ολική αναδημιουργία του προγράμματος.

Τέλος, η χρήση της defeasible logic που έχει χρησιμοποιηθεί για ενεργειακές αποφάσεις, θέτει ορισμένους περιορισμούς. Το κύριο πρόβλημα που προκύπτει είναι η έλλειψη προαπαιτούμενων στους κανόνες προτίμησης (preference). Οι συνθήκες αυτές διαδραματίζουν σημαντικό ρόλο στην διαμόρφωση του κώδικα τόσο από άποψη συνοχής όσο και απλότητας. Έτσι με την χρήση της defeasible logic απαιτούνται αυξημένοι πόροι και μεγάλος όγκος εντολών προκειμένου να επιτευχθεί το προσδοκώμενο αποτέλεσμα. Ακόμη όμως και σε αυτή την περίπτωση είναι πιθανόν να υπάρξουν συγκρούσεις μεταξύ διαφορετικών κανόνων που δεν είναι δυνατόν να επιλυθούν με αυτή την λογική.

Συνεπώς, προκειμένου να γίνει επίλυση των παραπάνω προβλημάτων σε σημαντικό βαθμό, έγινε χρήση ιεραρχικής επιχειρηματολογίας. Πάνω σε αυτή την λογική είναι βασισμένος και ο Gorgias τον οποίο αξιοποιήσαμε για την δημιουργία της εφαρμογής που παρουσιάζεται στην συνέχεια.

## Κεφάλαιο 2°

### Θεωρητικό υπόβαθρο

#### 2.1 Gorgias Framework

Ο Gorgias [5][6][7] είναι ένα πλαίσιο επιχειρηματολογίας το οποίο συνδυάζει την λογική της συλλογικής προτίμησης, της λογικής αλλά και της απαγωγής (abduction). Το framework αυτό είναι βασισμένο στην γλώσσα Prolog. Μπορεί να αποτελέσει την βάση για τη συλλογιστική, σχετικά με πολιτικές προτιμήσεων, έχοντας ελλείψεις πληροφορίες σε δυναμικά περιβάλλοντα.

Η χρήση επιχειρηματολογίας μπορεί να χρησιμοποιείται σε ένα ευρύ πλαίσιο προβλημάτων όπως την επίλυση νομικών ζητημάτων, την λήψη φαρμακευτικής αγωγής ακόμη και στο ηλεκτρονικό εμπόριο. Ο Gorgias ουσιαστικά κλείνει το κενό που υπάρχει στην ανάπτυξη ολοκληρωμένων εφαρμογών λήψης αποφάσεων με την χρήση επιχειρηματολογίας.

Ο πλαίσιο πάνω στο οποίο είναι βασισμένος ο Gorgias είναι αυτό της ιεραρχικής επιχειρηματολογίας (hierarchical argumentation). Η λήψη απόφασης με αυτή την μέθοδο προϋποθέτει την ύπαρξη ορισμένων βασικών στοιχείων:

- Ένα σύνολο επιλογών (options)
- Ένα σύνολο τιμών (values) που παραμετροποιούν τις επιλογές
- Επίπεδα επιχειρημάτων που περιλαμβάνουν: δομή, δηλαδή ένα σύνολο παραμέτρων που κάνουν μια επιλογή διαθέσιμη ή μη, για μια δεδομένη συνθήκη
- Προτεραιότητες οι οποίες δίνουν και την βαρύτητα στα διαφορετικά επιχειρήματα για διάφορες εναλλακτικές που υπάρχουν.

##### 2.1.1 Κανόνες του Gorgia

Ο Gorgias, προκειμένου λάβει μια απόφαση, είναι απαραίτητο να γραφτούν μια σειρά από κανόνες, οι οποίοι σχηματίζονται από κάποια σύμβολα κατηγοριοποίησης. Η φόρμα που ακολουθείται λοιπόν για την σύνταξη των κανόνων αυτών είναι: rule( Label, Head, Body). Το συμπέρασμα (Head) είναι ένα κατηγορήμα, το σώμα (Body) μια λίστα από κατηγορήματα και η ετικέτα (Label) είναι ένας σύνθετος όρος, που αποτελείται από ένα όνομα κανόνα και επιλεγμένες μεταβλητές από το κεφάλι και το σώμα. Για παράδειγμα, ο κανόνας rule(r1(x), fly(x), [bird(x)]) δηλώνει ότι κάτι πετάει εφόσον είναι πουλί.

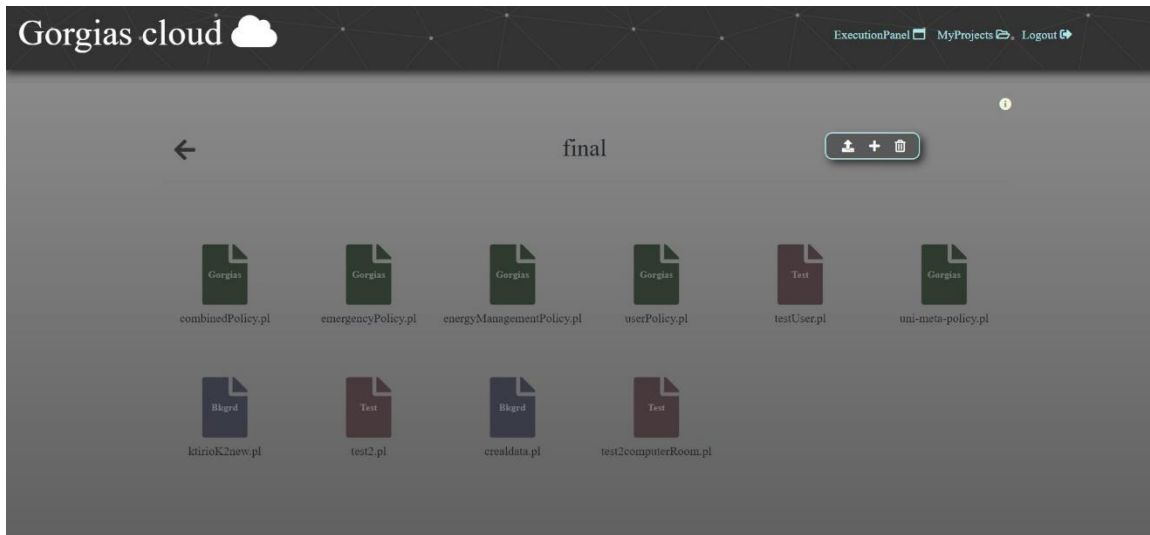
Επιπλέον, σημαντική είναι η εισαγωγή προτιμήσεων στους κανόνες, ώστε να γίνεται ξεκάθαρο ποιος κανόνας έχει την μεγαλύτερη βαρύτητα από έναν άλλον. Έτσι μπορούν να αποφευχθούν αντιπαραθέσεις (conflicts) μεταξύ διαφορετικών κανόνων. Ο τρόπος σύνταξης των προτιμήσεων είναι ο εξής: rule(Signature, prefer(Sig1, Sig2), Body). Αυτό σημαίνει ότι ο κανόνας με ταμπέλα

(label) Sig1, έχει μεγαλύτερη βαρύτητα από τον κανόνα Sig2, δεδομένων των προϋποθέσεων που έχουν τεθεί στο Body.

### 2.1.2 Gorgias Cloud

Αποτελεί την διαδικτυακή υπηρεσία του Gorgia για την λήψη αποφάσεων στο υπολογιστικό νέφος (cloud) [8]. Μέσω αυτής της υπηρεσίας η πρόσβαση στον Gorgia γίνεται ευκολότερη παρέχοντας ένα προσιτό και εύχρηστο περιβάλλον λειτουργίας. Με χρήση του cloud service είναι δυνατή η χρήση από οποιονδήποτε χρήστη, όπου και αν βρίσκεται, μόνο με την χρήση του διαδικτύου χωρίς να χρειάζεται να εγκαταστήσει την Prolog.

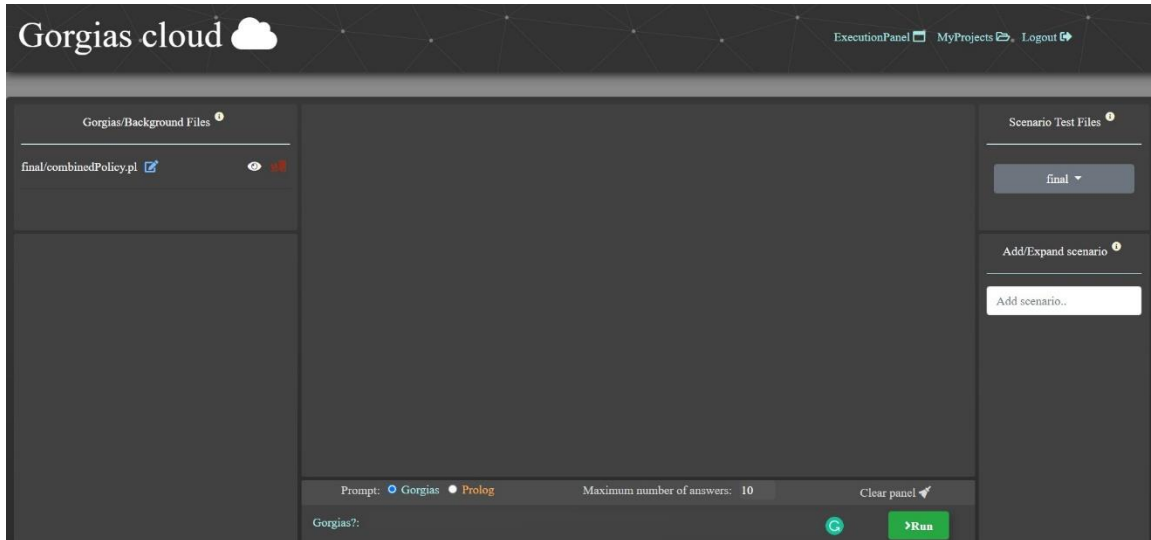
Χρησιμοποιώντας τις ίδιες αρχές και τους κανόνες σύνταξης του Gorgia, είναι δυνατόν ο χρήστης να δημιουργήσει ένα λογαριασμό και να “τρέξει” τα επιθυμητά σενάρια. Αρχικά ο χρήστης δημιουργεί ένα project μέσα στο οποίο τοποθετεί όλα τα αρχεία που θέλει να δημιουργήσει. Υπάρχουν τρεις τύποι αρχείων: Gorgias file, Background, Test scenario. Ο πρώτος τύπος εμπεριέχει το σύνολο των κανόνων που θέλουμε να εφαρμόσουμε για την λήψη μιας απόφασης. Ο δεύτερος τύπος περιέχει τις βασικές αρχές, οι οποίες παραμένουν σταθερές ανεξάρτητα από τους κανόνες που έχουν τεθεί. Ο τελευταίος τύπος αρχείου, Test scenario, εμπεριέχει ένα σύνολο παραμέτρων που έχουμε θέσει με σκοπό την δοκιμή κάποιου σεναρίου και την λήψη της απόφασης βάσει αυτών.



ΕΙΚΟΝΑ 1. Σύνολο Αρχείων στο Gorgias cloud

Σε επόμενο στάδιο αφού ο χρήστης έχει δημιουργήσει όλα τα αρχεία με τους κανόνες και τις παραμέτρους που τον ενδιαφέρουν, μπορεί να κάνει consult τα επιθυμητά αρχεία στο execution panel. Με αυτό τον τρόπο, χρησιμοποιώντας ένα αρχείο τύπου gorgias/background που εμπεριέχει τους κανόνες και ένα αρχείο τύπου scenario που εμπεριέχει τις βασικές παραμέτρους, μπορεί να γίνει η λήψη μιας απόφασης. Στο κάτω μέρος της οθόνης γίνεται η ερώτηση που μας ενδιαφέρει

και στην συνέχεια, το πρόγραμμα, συνδυάζοντας τα αρχεία που του έχουμε δώσει, δίνει την απάντηση.



ΕΙΚΟΝΑ 2. Το Execution panel του Gorgia

## 2.2 Πρωτόκολλο KNX

Το KNX [9] είναι ένα ανοιχτό πρότυπο επικοινωνίας που δημιουργήθηκε από την KNX association το 1990 στις Βρυξέλες. Πρόκειται για την ένωση τριών μεγάλων δικτύων διαχείρισης κατοικιών και επαγγελματικών κτιρίων στην ΕΕ (Ευρωπαϊκή Ένωση). Αυτοί οι οργανισμοί ήταν: HSA (European Home Systems Association), BCI (BatiBUS Club International) και EIBA (European Installation Bus Association. Μέσω αυτής της συνεργασίας προέκυψε το σύστημα EIB (European Installation Bus).

Το KNX διαθέτει πιστοποίηση EN 50090, ως Ευρωπαϊκό πρότυπο, και πιστοποίηση EN 50090, ως διεθνές πρότυπο βασισμένο στο μοντέλο OSI, που εστιάζει στον κτιριακό αυτοματισμό.

Ο κυρίαρχος στόχος δημιουργίας του KNX είναι η έμφαση στην άνεση και την ευχρηστία στην ζωή του σύγχρονου ανθρώπου. Μέσω του προτύπου, καθίσταται εύκολη η διασύνδεση μιας μεγάλης γκάμας διαφορετικών συσκευών. Αυτές οι συσκευές μπορούν να κατηγοριοποιηθούν σε πέντε βασικές κατηγορίες:

- **Δίαυλος επικοινωνίας (BUS)**

Ο διάυλος επικοινωνίας (BUS) αποτελεί για τον τρόπο σύνδεσης των διαφορετικών συσκευών μεταξύ τους. Όλες οι πληροφορίες και οι εντολές για τον έλεγχο της εγκατάστασης μεταδίδονται μέσω αυτού του διαύλου, όπου πάνω του συνδέονται όλες οι



ψηφιακές συσκευές του συστήματος. Τοποθετείται παράλληλα με την παροχή του ρεύματος σε όλες τις συσκευές και τα συστήματα στο δίκτυο.

- **Αισθητήρες (Sensors)**

Αποτελούν το μέσο επικοινωνίας ανάμεσα στο περιβάλλον και το σύστημα. Ανάλογα με το είδος αισθητήρα που είναι εγκατεστημένος είναι δυνατόν να λάβουμε και διαφορετικές τιμές από το περιβάλλον, οι οποίες δύναται να επεξεργαστούν από το σύστημα ώστε να πραγματοποιηθεί μια λειτουργία. Τέτοιες τιμές μπορεί να είναι η θερμοκρασία, η φωτεινότητα, η υγρασία, η ταχύτητα του ανέμου κ.α. Οι αισθητήρες λειτουργούν είτε αυτόνομα είτε βρίσκονται συνδεδεμένοι με τον δίαυλο επικοινωνίας, ώστε να συνδυάζονται οι μετρήσεις που λαμβάνουν.

- **Ενεργοποιητές (Actuators)**

Οι αισθητήρες, μέσω των μετρήσεων που λαμβάνουν και αφού αυτές επεξεργαστούν, δίνουν εντολή στον ενεργοποιητή για εκτέλεση των εντολών. Αυτές οι συσκευές διανέμουν ισχύ στα διάφορα ηλεκτρικά κυκλώματα με τα οποία είναι συνδεδεμένη η μονάδα ισχύος τους. Το διαδεδομένο είδος ενεργοποιητών είναι οι διακόπτες On/Off (Switch Actuator) που προγραμματίζονται για το άνοιγμα-κλείσιμο συσκευών οι οποίες είναι συνδεδεμένες σε αυτούς,

- **Ελεγκτές (Controllers)**

Πολλοί ενεργοποιητές αλλά και αισθητήρες είναι πιθανόν να διαθέτουν ενσωματωμένο ελεγκτή προκειμένου να επικοινωνήσουν μέσω του BUS και να πραγματοποιηθεί απευθείας η ζητούμενη λειτουργία.

Συμπερασματικά, μπορούμε να πούμε πως το πρότυπο KNX αποτελεί το κυρίαρχο πρότυπο επικοινωνίας συσκευών με πάνω από 419 κατασκευαστές εξοπλισμού ανάμεσα τους εταιρίες όπως ABB και Siemens, που έχουν σχεδιάσει και προωθούν στην αγορά μια μεγάλη ποικιλία υποστηριζόμενων συσκευών.

## 2.3 Διεπαφή Προγραμματισμού Εφαρμογών (API)

Η Διεπαφή Προγραμματισμού Εφαρμογών (Application Programming Interface -API) αποτελεί ένα μέσο περιγραφής χαρακτηριστικών μιας βιβλιοθήκης ή την επικοινωνία με αυτή. Πρόκειται για την διεπαφή προγραμματιστικών διαδικασιών μεταξύ ενός λειτουργικού συστήματος, μιας εφαρμογής ή μια βιβλιοθήκης προκειμένου να γίνουν αιτήσεις ή ανταλλαγή δεδομένων από άλλα προγράμματα. Κάθε εφαρμογή, προκειμένου να κάνει χρήση του API, είναι απαραίτητο να διαθέτει ένα “API Documentation”, το οποίο αποτελεί ένα εγχειρίδιο λειτουργίας όπου ο προγραμματιστής μπορεί να δει ποιες συναρτήσεις είναι διαθέσιμες και πώς μπορεί να τις καλέσει.

Κάθε διαδικτυακή υπηρεσία διαθέτει, εξ ορισμού, έναν τρόπο επικοινωνίας συνήθως με την χρήση προτύπων XML ή JSON. Μέσω αυτών μπορεί ο χρήστης να καλέσει την διαδικτυακή υπηρεσία που επιθυμεί, με την αποστολή ενός αιτήματος και λαμβάνοντας ως ανταπόκριση μια απάντηση. Συνήθως, όταν αυτή η επικοινωνία πραγματοποιείται στον χώρο του διαδικτύου, γίνεται μέσω του πρωτοκόλλου HTTP που αποτελεί και το δημοφιλές πρωτόκολλο για αυτού του είδους την επικοινωνία.

Τόσο λοιπόν η διαδικτυακή υπηρεσία, όσο και ένα API, αποτελούν έναν τρόπο επικοινωνίας μεταξύ μηχανών. Βέβαια η διαφορά τους έγκειται στο γεγονός ότι η διαδικτυακή υπηρεσία έχει ως μέσο επικοινωνίας το διαδίκτυο. Σε αντίθεση, το API καθορίζει τις μεθόδους επικοινωνίας μεταξύ των μηχανών, λειτουργώντας ως διεπαφή. Όταν αυτή η διεπαφή γίνεται μέσω διαδικτύου μιλάμε για μια διαδικτυακή πλέον υπηρεσία. Καταλήγοντας, μπορούμε να βγάλουμε το συμπέρασμα ότι κάθε διαδικτυακή υπηρεσία αποτελεί μια διεπαφή χωρίς όμως να ισχύει αναγκαία και το αντίστροφο.

## 2.4 Representational State Transfer (REST)

Το Representational State Transfer (REST) [10], αποτελεί μια αρχιτεκτονική για σχεδιασμό διαδικτυακών εφαρμογών στον Παγκόσμιο Ιστό (World Wide Web-WWW). Το REST περιλαμβάνει ένα σύνολο προϋποθέσεων σχετικά με το πώς ένα διαδικτυακό σύστημα συμπεριφέρεται. Το REST χρησιμοποιεί το πρωτόκολλο HTTP για τις κλήσεις μεταξύ των μηχανών κάτι που το κάνει ιδιαίτερος πρακτικό αφού είναι βασισμένο στο ίδιο πρωτόκολλο με το World Wide Web. Επιπρόσθετα είναι ιδιαίτερα “ελαφρύ” με την έννοια του ότι απαιτεί λιγότερους διαθέσιμους πόρους σε αντίθεση με άλλους αντίστοιχους μηχανισμούς όπως το RPC (Remote Procedure Calls) ή ηλεκτρονικές υπηρεσίες όπως τα SOAP, WSDL [10].

Στην παρούσα διπλωματική χρησιμοποιήθηκε η αρχιτεκτονική REST, καθώς υπήρχε η απαραίτητη υποστήριξη μέσω εγχειριδίων (Documentation) που περιείχαν τις κατάλληλες μορφές δεδομένων. Συγκεκριμένα χρησιμοποιήθηκε για την σύνδεση των αισθητήρων και λήψη δεδομένων εξ αυτών αλλά και για την επικοινωνία του Gorgia Cloud με το σύστημα που αναπτύχθηκε για την αποστολή δεδομένων και την λήψη αποφάσεων.

## 2.5 Swagger

Το Swagger [11] αποτελεί ένα περιβάλλον διεπαφής που στόχο έχει την παραγωγή βιβλιοθηκών πελάτη. Μέσω της online πλατφόρμας είναι δυνατόν να δεχθεί ως είσοδο ένα σύνολο διαφορετικών αρχείων όπως: JSON και OpenAPI. Μέσω αυτού του αρχείου μπορεί να παραχθεί η βιβλιοθήκη σε 34 διαφορετικές γλώσσες προγραμματισμού. Με αυτό τον τρόπο λοιπόν είναι δυνατή η παραγωγή της απαιτούμενης βιβλιοθήκης, χωρίς να χρειάζεται η αποστολή ξεχωριστών αιτημάτων για κάθε πόρο, για το σύνολο του API. Στην συγκεκριμένη εργασία το Swagger χρησιμοποιήθηκε

για την βιβλιοθήκη του REST API του Gorgias Cloud, χρησιμοποιώντας την γλώσσα προγραμματισμού Python.

## 2.6 PyCharm

Το PyCharm [12] αποτελεί ένα Ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για χρήση σε προγραμματισμό υπολογιστών. Είναι φτιαγμένο ειδικά για την γλώσσα python. Έχει αναπτυχθεί από την Τσέχικη εταιρία JetBrains. Υποστηρίζει μια πολύ μεγάλη ποικιλία βιβλιοθηκών για την εύκολη ανάπτυξη προγραμμάτων. Το PyCharm παρέχει ανάλυση κώδικα, πρόγραμμα εντοπισμού σφαλμάτων, μονάδα δοκιμής κώδικα καθώς και απασφαλτωτή (debugger). Τέλος ένα από τα κύρια χαρακτηριστικά τα οποία διαθέτει είναι η υποστήριξη στην ανάπτυξη διαδικτυακών εφαρμογών όπως του Django, το οποίο χρησιμοποιήθηκε και στην παρούσα εργασία.

## 2.7 Django

Το Django [13] είναι ένα ελεύθερο πλαίσιο ανάπτυξης διαδικτυακών εφαρμογών (WAF), υψηλού επιπέδου (high level) βασισμένο σε Python. Δημιουργήθηκε το 2003 από μια ομάδα προγραμματιστών που χρησιμοποιούσαν Python για την δημιουργία των εφαρμογών τους. Η πρώτη έκδοση έγινε δύο χρόνια αργότερα, το 2005, υπό την άδεια BSD. Η ονομασία του εργαλείου αποδίδεται στον κιθαρίστα Django Reinhardt. Ακολουθεί το αρχιτεκτονικό μοτίβο Model Template Views (MTV) ή την Model View Controller (MVC). Συγκεκριμένα τα τρία μέρη που αποτελείται είναι τα εξής:

- **Model**  
Αναφέρεται στην αναπαράσταση και στην συσχέτιση των δεδομένων που ανταλλάσσονται μεταξύ του διακομιστή και της βάσης δεδομένων. Συγκεκριμένα γίνεται χρήση της τεχνικής Object Relational Mapper (ORM) σύμφωνα με την οποία μετατρέπει τα δεδομένα, με την χρήση αντικειμενοστραφών γλωσσών προγραμματισμού, μεταξύ δύο μη συμβατών συστημάτων.
- **View**  
Πρόκειται για την αναπαράσταση των δεδομένων στον χρήστη. Αυτή η διαδικασία γίνεται μέσω μιας σελίδας HTML μέσω της οποίας μπορεί ο χρήστης να αλληλοεπιδρά με το πρόγραμμα περιήγησης.
- **Controller**  
Ο controller είναι το πλέον βασικό κομμάτι του συστήματος. Είναι υπεύθυνος για την ροή ελέγχου της πληροφορίας, δηλαδή από την βάση δεδομένων μέχρι την εμφάνιση της

πληροφορίας στην οθόνη του χρήστη. Επιπλέον μπορεί να γίνει η διαχείριση Ερωτημάτων και Απαντήσεων (Requests and Responses).

Ο κύριος στόχος υλοποίησης προγραμμάτων σε Django είναι η αποφυγή δημιουργίας σύνθετων ιστοσελίδων που χρησιμοποιούν βάσεις δεδομένων. Επιπλέον δίνεται μεγάλη έμφαση στην ταχύτητα ανάπτυξης, την επαναχρησιμοποίηση των συστατικών της και στην αρχή DRY (don't repeat yourself).

Συνοπτικά μπορούμε να κατηγοριοποιήσουμε τα χαρακτηριστικά του Django στα παρακάτω:

- **ORM (Object-relational mapper)**

Αναφέρονται στα μοντέλα της εφαρμογής όπου είναι γραμμένα αποκλειστικά σε Python. Με αυτό τον τρόπο αποφεύγεται η ανάγκη να γραφτεί SQL για τα queries που γίνονται στην βάση. Παρόλα αυτά ο προγραμματιστής δύναται να χρησιμοποιήσει SQL για περίπλοκα ερωτήματα προς τη βάση.

- **Αυτόματη διεπαφή διαχειριστή**

Το διαχειριστικό περιβάλλον την εφαρμογής είναι ενσωματωμένο στο Django. Έτσι ο προγραμματιστής δεν χρειάζεται να δημιουργήσει από το μηδέν το σύστημα διαχείρισης των χρηστών, των ρόλων και των περιεχομένων.

- **Κρυφή μνήμη (Cache)**

Η κρυφή μνήμη αποθηκεύει ένα σύνολο υπολογισμών που έχουν πραγματοποιηθεί όταν ο χρήστης κάνει Request μιας σελίδας από τον Web server. Με αυτό τον τρόπο αποφεύγονται οι επιπλέον υπολογισμοί και η επανεκτέλεση αυτών, επιστρέφοντας την επιθυμητή απάντηση στον χρήστη σε μικρό χρονικό διάστημα.

- **Σχεδιασμός URL**

Ο σχεδιασμός του URL παίζει σημαντικό ρόλο για το σύνολο της εφαρμογής όπως είναι η ασφάλεια και η ποιότητα της. Ο τρόπος που χειρίζεται το Django τα URL, το καθιστά εύχρηστο και ευέλικτο.

- **Template σύστημα**

Προκειμένου να γίνεται πιο εύκολος ο σχεδιασμός, το Django διαθέτει μια μεγάλη γκάμα από templates. Συγκεκριμένα, μέσω της template language καθίσταται εύκολη η διαχείριση της πληροφορίας στις σελίδες.

- **Ασφαλές**

Προσφέρει στον προγραμματιστή την δυνατότητα χρήσης του προτύπου HTTPS στις ιστοσελίδες που εφαρμόζεται. Επιπλέον τα templates που διαθέτει, ενσωματώνουν χαρακτήρες διαφυγής (Escape Specific Characters), μέσω των οποίων ο χρήστης είναι προστατευμένος από XSS attack (Cross Site Scripting). Τέλος διαθέτει την κατάλληλη τεχνολογία ώστε να προστατεύει τους χρήστες από το SQL Injection, μιας τεχνικής ιδιαίτερα διαδεδομένης για επιθέσεις σε εφαρμογές που διαχειρίζονται δεδομένα.

- **Επεκτάσιμο**

Μπορεί να δεχθεί πληθώρα προσθηκών που έχουν σχεδιαστεί από εξωτερικούς προγραμματιστές (Third party). Η ενσωμάτωση αυτών μπορεί να γίνει με πολύ εύκολο

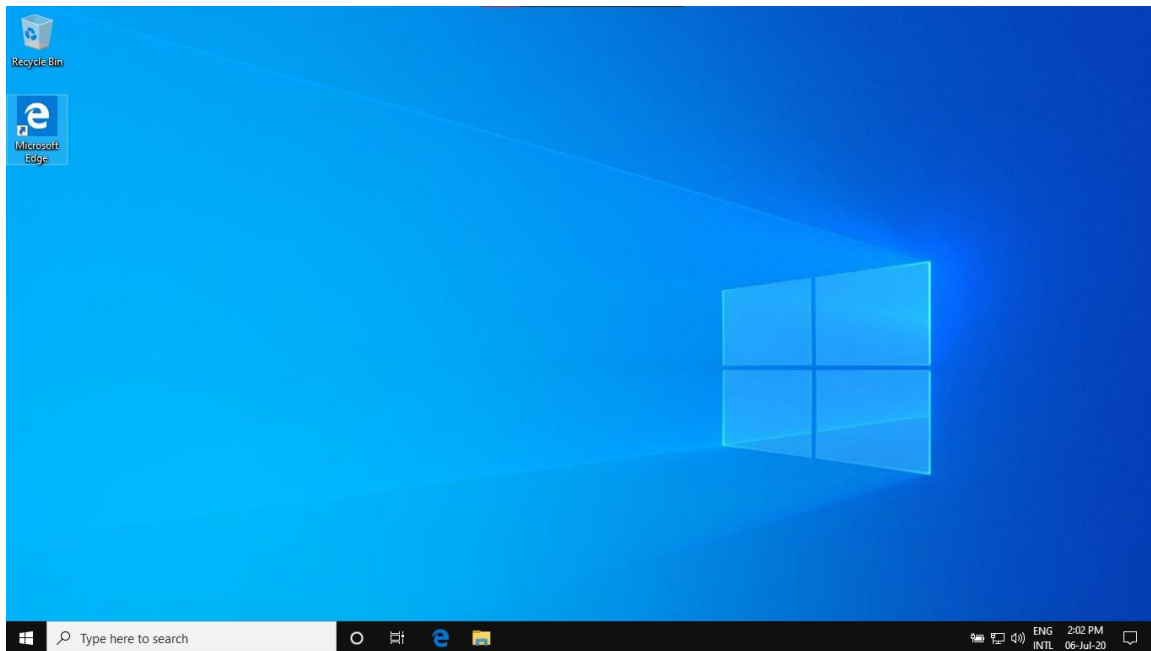
τρόπο, ενώ υπάρχουν περισσότερα από 2500 “πακέτα” τέτοιων εφαρμογών. Με την χρήση αυτών, επεκτείνονται οι δυνατότητες του Framework, ενώ δίνονται λύσεις σε προβλήματα που πιθανώς να υπάρξουν.

Τέλος, αξίζει να σημειωθεί ότι μια σειρά από μεγάλους ιστότοπους, παγκοσμίως, χρησιμοποιούν το συγκεκριμένο Framework, όπως: ο Mozilla Firefox , το Instagram, το Pinterest και το National Geographic.

## 2.8 Γραφικό Περιβάλλον Χρήστη- Graphic User Interface

Το Γραφικό Περιβάλλον Χρήστη (GUI) είναι το σύνολο εικονικών στοιχείων (γραφικών), που παρουσιάζονται στην οθόνη μιας συσκευής όπως ενός ηλεκτρονικού υπολογιστή. Μέσω αυτού του περιβάλλοντος ο χρήστης μπορεί να αλληλεπιδρά με το σύστημα για την πραγματοποίηση ορισμένων διαδικασιών-εργασιών. Αυτή η διεπαφή γίνεται συνήθως με την χρήση κάποιας συσκευής εισόδου όπως το πληκτρολόγιο ή το ποντίκι. Το περιβάλλον αυτό μπορεί να περιέχει σύμβολα, εικονίδιο, κουμπιά όπως και ένα σύνολο γραφικών στοιχείων, τα οποία διευκολύνουν τον χρήστη προσεγγίζοντας, κατά το δυνατόν, την εμπειρία στο φυσικό περιβάλλον. Με αυτό τον τρόπο, δημιουργείται ένα γραφικό περιβάλλον στον υπολογιστή, το οποίο είναι εύχρηστο και λειτουργικό.

Ένα από τα πιο συνηθισμένα παραδείγματα GUI αποτελεί το λειτουργικό σύστημα του υπολογιστή μας. Παρακάτω παρουσιάζεται η επιφάνεια εργασίας των Windows 10, όπου είναι ευδιάκριτα τα χαρακτηριστικά που αποτελούν το GUI όπως τα εικονίδια, η γραμμή ενημερώσεων, το κουμπί εκκίνησης, ανάμεσα στα οποία ο χρήστης μπορεί να περιηγηθεί χρησιμοποιώντας το ποντίκι.



ΕΙΚΟΝΑ 3. Το γραφικό περιβάλλον των windows 10

## 2.9 HTML forms

Αποτελούν ένα βασικό κομμάτι του Γραφικού Περιβάλλοντος Χρήστη (Graphic User Interface) αναφορικά με την εμφάνιση. Μέσω αυτής της φόρμας, ο χρήστης της ιστοσελίδας έχει την δυνατότητα να εισάγει δεδομένα, τα οποία στέλνονται στον εξυπηρετητή για περαιτέρω επεξεργασία. Οι φόρμες συνήθως περιέχουν πεδία-κελία τα οποία καλείται να συμπληρώσει ο χρήστης, προκειμένου να αναζητήσουν κάποια πληροφορία ή να απαντήσουν σε ερωτήσεις πολλαπλών επιλογών. Τα στοιχεία εισόδου σε μια φόρμα είναι τα παρακάτω:

- **Πεδίο εισόδου (inputs)**  
Πρόκειται για το πεδίο στο οποίο ο χρήστης εισάγει την πληροφορία με έναν σύντομο τρόπο, επιλέγοντας ανάμεσα σε κείμενο μιας γραμμής (text input), είτε σε ένα κουτί επιλογών (check box), είτε σε ένα κουτί πολλαπλών επιλογών (radio box). Επιπλέον σε αυτό το σημείο μπορεί να γίνει “ανέβασμα” κάποιου αρχείου. Τέλος υπάρχει και η επιλογή «αποστολή» (submit) η οποία καλεί τον φυλλομετρητή να εκτελέσει μια ενέργεια, όπως είναι η αποστολή των δεδομένων που έχουν εισαχθεί στα πεδία.
- **Περιοχή κειμένου (text area)**  
Αυτό το πεδίο μοιάζει με το text input, αλλά διαφοροποιείται με το μέγεθος του κειμένου που εισάγει ο χρήστης. Σε αυτό το σημείο είναι δυνατόν να εισαχθούν πολλαπλές γραμμές κειμένου.
- **Επιλογή λίστας (drop-down list)**

Παρουσιάζεται μια λίστα που περιέχεται ένα σύνολο διαφορετικών στοιχείων-δεδομένων ανάμεσα στα οποία μπορεί ο χρήστης να επιλέξει.

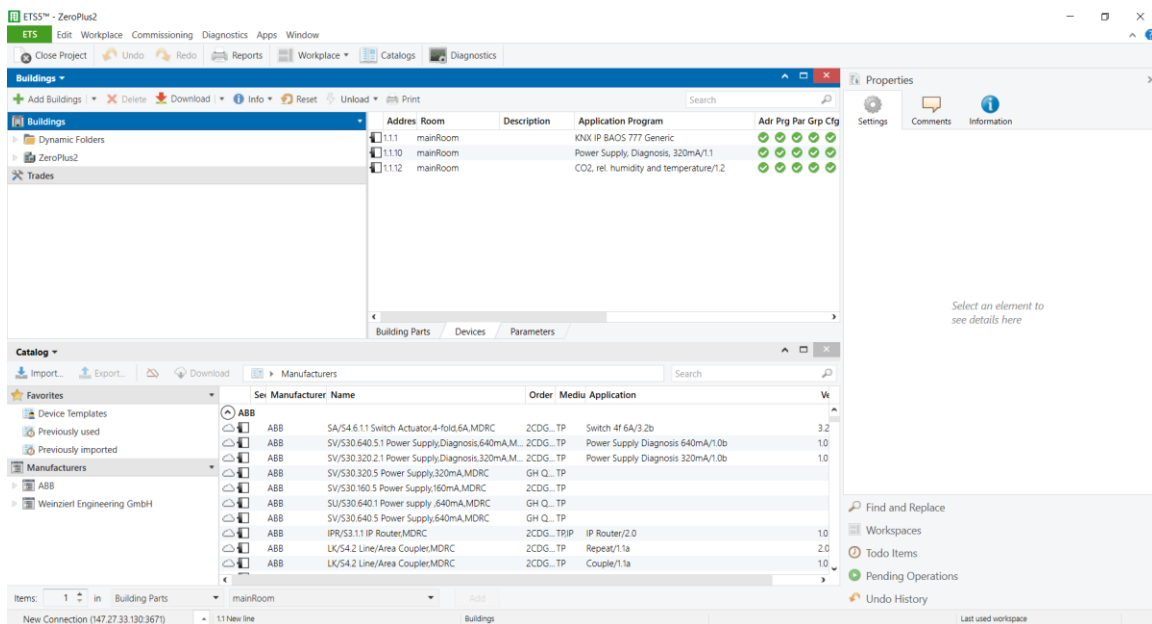
## 2.10 Engineering Tool Software 5

Το ETS5 [14] (Engineering Tool Software 5) είναι το πρότυπο εργαλείο που χρησιμοποιείται για την διασύνδεση συσκευών που βασίζονται στο πρότυπο KNX. Συνολικά υποστηρίζονται περισσότερες από 7.000 πιστοποιημένες συσκευές από 350 διαφορετικούς κατασκευαστές. Μέσω του εργαλείου είναι δυνατή η διασύνδεση και η ενσωμάτωση στο σύστημα όλων των συμβατών συσκευών ακόμη και αυτών που διαθέτουν τεχνολογία ασύρματης σύνδεσης. Μέσω project που δημιουργεί ο χρήστης, μπορεί να γίνει η αποτύπωση των χώρων που είναι διαθέσιμοι όπως σαλόνι, κουζίνα, υπνοδωμάτιο ακόμη και διαφορετικοί όροφοι. Στην συνέχεια, μέσω λίστας, γίνεται εύρεση των αισθητήρων που είναι διαθέσιμοι καθώς και η διασύνδεση αυτών μέσα στο συγκεκριμένο project. Έπειτα ο χρήστης ταξινομεί τους αισθητήρες και τους ενεργοποιητές στους διαφορετικούς χώρους που έχει δημιουργήσει. Μέσω αυτής της ομαδοποίησης γίνεται ευκολότερη και η αλληλεπίδραση των διαφορετικών συσκευών.

Το παρόν πρόγραμμα διαθέτει 4 διαφορετικές εκδόσεις ανάλογα με τον αριθμό των συσκευών που επιθυμεί ο χρήστης να χρησιμοποιήσει, τα πιθανά σενάρια που πρέπει να εκτελεστούν αλλά και τον αριθμό των διαφορετικών υπολογιστών που διαθέτουν άδεια.

Στην παρούσα εργασία, προκειμένου να προγραμματίσουμε τους αισθητήρες που διαθέτουμε, έγινε η χρήση του ETS5 στην έκδοση Demo, η οποία επιτρέπει κατά μέγιστο 3 συσκευές που ήταν και ο αριθμός που διαθέτουμε. Συγκεκριμένα, μέσα από το menu του συστήματος βρέθηκαν, σε πρώτη φάση, οι 3 αισθητήρες που έχουμε στην κατοχή μας και τοποθετήθηκαν σε ένα κοινό αρχείο ίδιας ομάδας καθώς μας ενδιαφέρει η αλληλεπίδραση μεταξύ αυτών. Έπειτα έγινε ο συγχρονισμός των αισθητήρων, κατεβάζοντας και φορτώνοντας τα δεδομένα στις συσκευές με ασύρματο τρόπο.

Με την ολοκλήρωση αυτής της διαδικασίας οι αισθητήρες είναι έτοιμοι για χρήση, ώστε να μας τροφοδοτήσουν με τα αναγκαία δεδομένα όπως θερμοκρασία, υγρασία, κατανάλωση ρεύματος τα οποία στην συνέχεια χρησιμοποιήθηκαν για την λήψη αποφάσεων στο κτίριο.

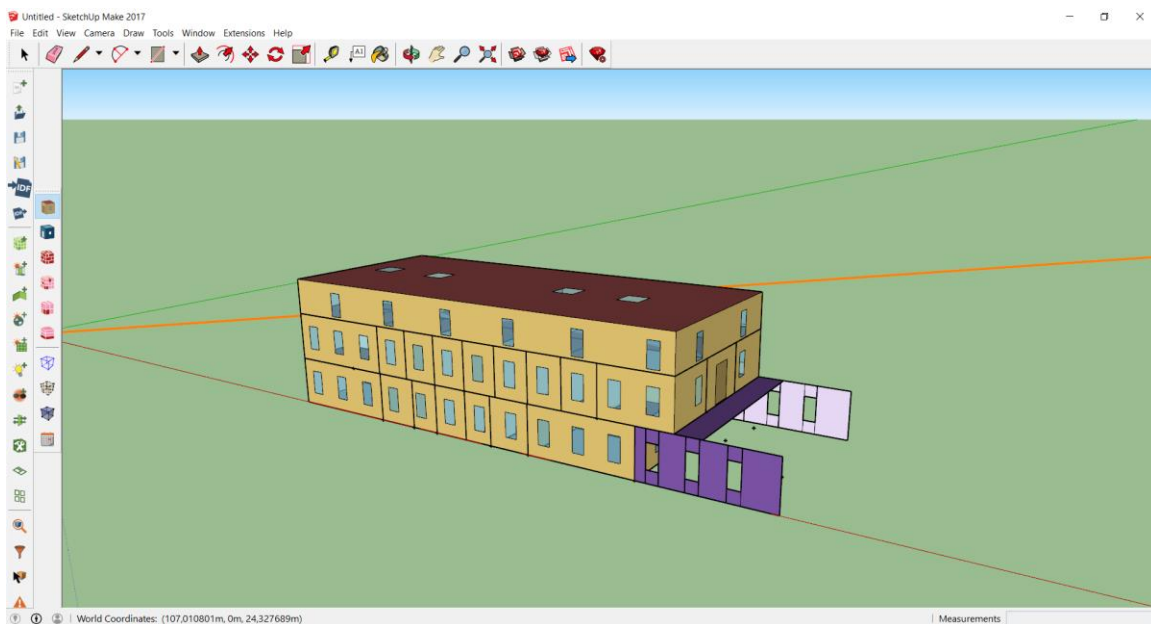


ΕΙΚΟΝΑ 4. Η βασική οθόνη επιλογών του ETS5

## 2.11 Sketch Up- Open studio

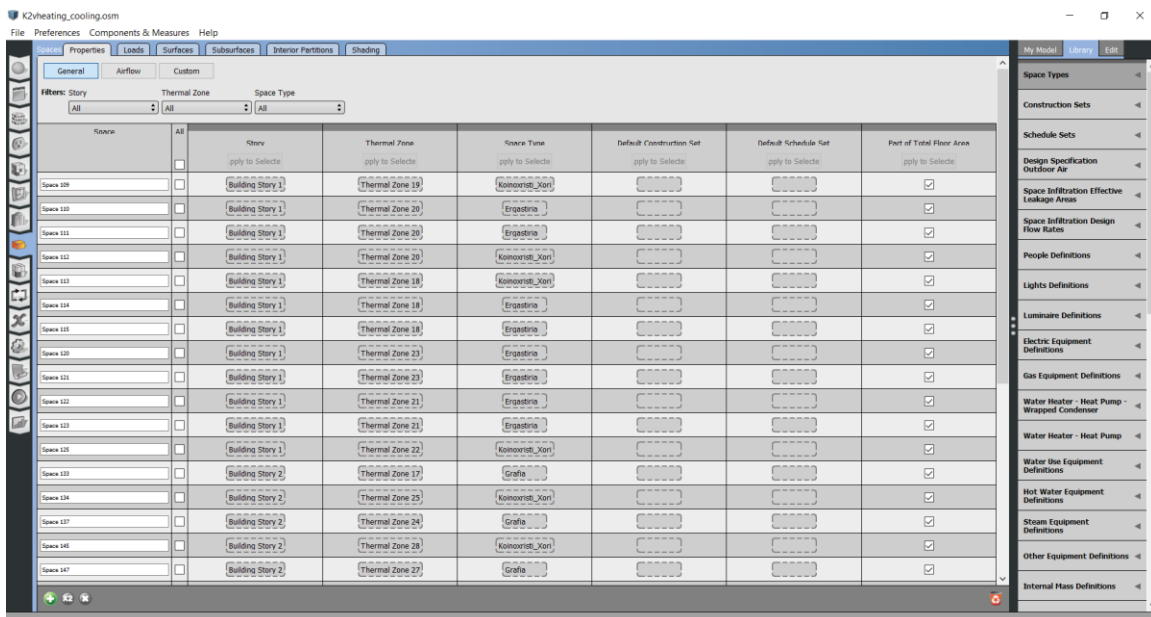
Το Sketch Up αποτελεί ένα λογισμικό τρισδιάστατης μοντελοποίησης κτιρίων αλλά και κάθε είδους εξωτερικού και εσωτερικού χώρου. Οι κύριοι χρήστες του προγράμματος είναι αρχιτέκτονες, πολιτικοί και μηχανολόγοι μηχανικοί. Λόγω βέβαια του φιλικού προς το χρήστη περιβάλλοντος, έχει χρησιμοποιηθεί από ένα μεγάλο εύρος χρηστών. Διαθέτει μία ευρεία ποικιλία εργαλείων, με τα οποία, ανάλογα με τις γνώσεις του χρήστη, είναι δυνατόν να σχεδιαστεί το υπό μελέτη μοντέλο με μία σχετική ακρίβεια έως και πολύ λεπτομερώς.





ΕΙΚΟΝΑ 5. Το κτίριο K2 μοντελοποιημένο στο Sketch Up

Το Open Studio αποτελεί μια επέκταση (plug-in) του προγράμματος Sketch-up και έχει αναπτυχθεί από το Εθνικό Εργαστήριο Ανανεώσιμων Πηγών των ΗΠΙΑ (NREL - National Renewable Energy Laboratory). Μέσω αυτού του λογισμικού είναι δυνατόν να γίνει ενεργειακή μοντελοποίηση του κτιρίου που έχει σχεδιαστεί σε προηγούμενο στάδιο, με την χρήση των εργαλείων που παρέχονται.



ΕΙΚΟΝΑ 6. Η οθόνη επιλογών του Open Studio

## Κεφάλαιο 3<sup>ο</sup>

### Προβληματική-Στόχοι εργασίας

#### 3.1 Ο Γοργίας

Ο Γοργίας ο Λεοντίνος [15] (483 -375 π.χ.) ήταν αρχαίος Έλληνας ρήτορας (σύγχρονος του Πρωταγόρα και του Σωκράτη) που ανήκε στους “προσωκρατικούς”, δηλαδή στο κίνημα των “σοφιστών”. Είναι διάσημος ο ομώνυμος, Σωκρατικός διάλογος που έγραψε ο Πλάτωνας περίπου το 380 π.χ. που διεξάγεται μεταξύ του Σωκράτη και μιας ομάδας σοφιστών περί της “ρητορείας”, η οποία βέβαια βασίζεται σε επιχειρήματα και λογική. Περιπλανώμενος, ο Γοργίας ζει στην Αθήνα στην περίοδο κορύφωσης της δόξας του, ενώ τα τελευταία χρόνια της ζωής του έζησε στη Θεσσαλία, όπου πέθανε σε βαθύ γήρας σε πλήρη πνευματική διαύγεια. Η επιλογή του ονόματος αυτού του προγράμματος λήψης αποφάσεων δεν είναι τυχαία, καθώς οι σοφιστές αρνούσαν την ύπαρξη της απόλυτης αλήθειας και πρόσβευαν ότι δεν υπάρχει “απόλυτη γνώση” αλλά μόνο “δόξα” δηλαδή “γνώμη” για την πραγματικότητα. Δεινοί ρήτορες, έπειθαν τους ακροατές τους επί παντός επιστητού. Η σύγχρονη λήψη αποφάσεων με την τεχνητή νοημοσύνη, βασίζεται στην επεξεργασία τεράστιου όγκου δεδομένων. Τα τρομακτικά αυτά μεγέθη των πληροφοριών, σίγουρα δεν αποδίδουν την “απόλυτη αλήθεια”, η οποία, κατά τους φιλοσόφους του νου, είναι χαρακτηριστικό του ανθρώπινου πνεύματος που αναζητά την μοναδική αλήθεια και τον Θεό. Παρόλα αυτά, προγράμματα όπως ο Γοργίας, μας βοηθούν να πλησιάσουμε το βέλτιστο αποτέλεσμα, που για την ειρωνεία του πράγματος, είναι μάλλον αντικειμενικό, μοναδικό και σίγουρα όχι “σόφισμα”. Γιατί, στις περισσότερες περιπτώσεις, εκεί τουλάχιστον που τα αριθμητικά – ποσοτικά δεδομένα είναι καθοριστικά, η επεξεργασία τεράστιων όγκων δεδομένων ξεπερνά τις ανθρώπινες δυνατότητες λήψης αποφάσεων.

#### 3.2 Γενικοί στόχοι

Ο προβληματισμός του θέματος προέρχεται από την σύγχρονη απαίτηση και αναζήτηση για την λήψη αποφάσεων μέσω τεχνητής νοημοσύνης. Ο μεγάλος όγκος πληροφοριών που έχουν τη δυνατότητα να επεξεργαστούν τα ηλεκτρονικά συστήματα, ανοίγει νέους δρόμους για την λήψη των βέλτιστων μέτρων και πρακτικών. Η ιατρική διάγνωση είναι ο πιο γνωστός τομέας που χαρακτηρίζεται και επωφελείται από την τεχνητή νοημοσύνη στη λήψη αποφάσεων.

Τα “έξυπνα κτίρια” αποτελούν άλλον ένα πολύ πρόσφορο χώρο για την εφαρμογή της λήψης αποφάσεων. Ο συγκεκριμένος χώρος παρέχει ένα μεγάλο εύρος εφαρμογών: ξεκινά από τη διαχείριση ενός κτιρίου με τον πιο οικονομικό, οικολογικά φιλικό τρόπο. Παράλληλα, ανοίγει ένα παράθυρο για την υλοποίηση του “ονείρου” του σύγχρονου ανθρώπου για ένα κτίριο που ικανοποιεί αλλά και προλαβαίνει τις καθημερινές ανάγκες του και ακόμη, τον φέρνει ένα βήμα πιο

κοντά στα σενάρια “επιστημονικής φαντασίας”. Επίσης, η δυνατότητα αυτή του παρέχεται εξ αποστάσεως, προλαβαίνοντας την άφιξη του ανθρώπου ή λειτουργούν κατά την απουσία του.

Τέλος, οι εφαρμογές που απορρέουν από ένα τέτοιο σύστημα, ξεφεύγουν από τη διαχείριση ενός απλού κτιρίου, καθώς χρησιμοποιούν ή εφαρμόζονται σε οικολογική βάση, με τεράστια δυνητικά οφέλη προς αυτή την κατεύθυνση. Η έρευνα που χρησιμοποιούν μπορεί να χρησιμοποιηθεί σε οικολογικούς σκοπούς για το μεγάλο “σπίτι” μας, τον πλανήτη.

Τα κτίρια του Πολυτεχνείου Κρήτης θέλουν να πρωτοπορήσουν, στο χώρο των Πανεπιστημιακών Ιδρυμάτων, για μια έξυπνη διαχείριση των τεράστιων δεδομένων που μπορούν να συλλεχθούν από περιβαλλοντικούς αισθητήρες και να αξιοποιηθούν.

Όπως φαίνεται παρακάτω, στα σενάρια διαχείρισης του gorgia (βλ. 5.5.), προσφέρονται λύσεις λειτουργικές σε θέματα όπως η εξοικονόμηση ενέργειας π.χ. με αισθητήρες που μετρούν το φάσμα επιθυμητών θερμοκρασιών στο χώρο και αυτορυθμίζονται. Όχι μόνο γίνεται αισθητή οικονομία πόρων, αλλά το πιο σημαντικό, παρέχεται υψηλή ποιότητα υπηρεσιών και διαβίωσης (όταν για παράδειγμα, το αμφιθέατρο θερμαίνεται κατά τις εργάσιμες ώρες, αυτόματα). (βλ. Σενάριο 8, 5.5.). Οι φοιτητές, αλλά και οι διδάσκοντες και το προσωπικό, επωφελούνται από την ασφάλεια για την διαφύλαξη της ανθρώπινης υγείας και ζωής που εξασφαλίζεται π.χ. από υψηλά επίπεδα διοξειδίου του άνθρακα και την ενεργοποίηση μηχανισμών alarm, π.χ. βλ. Σενάριο 3.

### 3.3 Περιθώρια εξέλιξης

Οι εφαρμογές του συστήματος είναι ανεξάντλητες, όσες και οι αναγκαιότητες που προκύπτουν στο παρόν και στο μέλλον. Τα έξυπνα γραφεία που ανιχνεύουν την ανθρώπινη παρουσία ή τα έξυπνα εργαστήρια ή οι χώροι σίτισης, προφυλάσσουν τον ακριβό εξοπλισμό από ακραίες ή αιφνίδιες μεταβολές του περιβάλλοντος, εξωτερικές ή εσωτερικές. Μια ευρείας κλίμακας εγκατάσταση τέτοιων συστημάτων, με έμφαση στην πρόληψη και την ασφάλεια, μπορούν να είναι εγγυητές ενός ιδανικού περιβάλλοντος εργασίας.

### 3.4 Δικλείδες ασφαλείας

Τα συστήματα τέτοιου τύπου μπορεί να είναι καθοριστικά ευεργετικά για τον χώρο του Πολυτεχνείου. Ωστόσο, αξίζει να λάβουμε υπόψη μας κάποιες αναγκαίες δικλείδες ασφαλείας που θα έχουν τους εξής στόχους: 1. διασφάλιση της ικανότητας αυτενέργειας του ανθρώπινου παράγοντα για λόγους ασφαλείας ή απλώς για λόγους ελευθερίας βούλησης σε ειδικές περιπτώσεις. 2. Καταγραφή προτύπων (standards) ηθικής που να διασφαλίζουν την ασφαλή χρήση τους, κατά το πρότυπο του “IEEE – Ethics in Artificial Intelligence” και των αντίστοιχων προτύπων που σχεδιάζονται από το Ινστιτούτο αυτό.

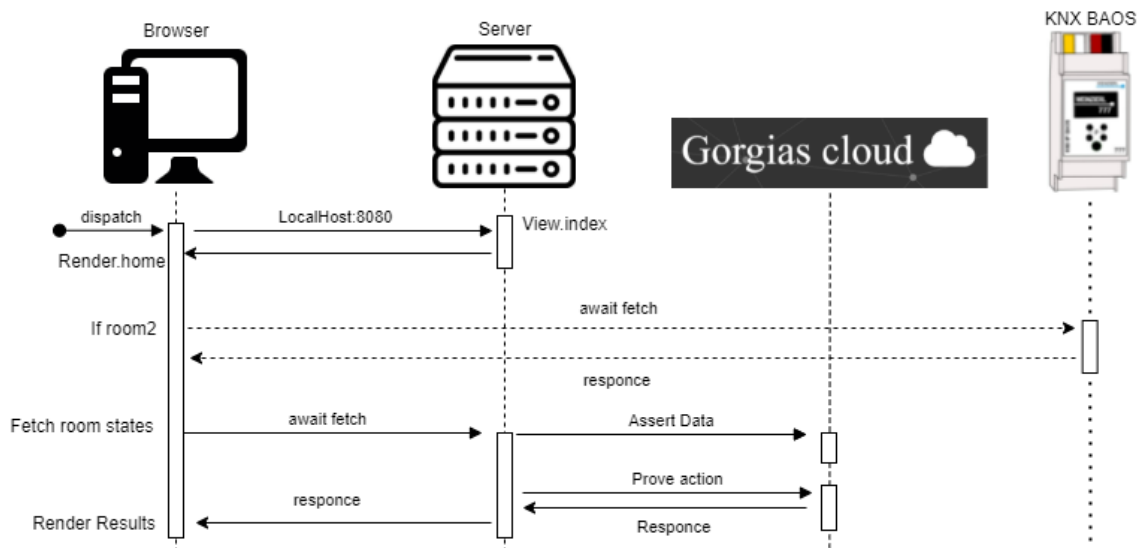
## Κεφάλαιο 4<sup>ο</sup>

### Ανάπτυξη εφαρμογής

#### 4.1 Αρχιτεκτονική

Η εφαρμογή, προκειμένου να αναπτυχθεί, απαιτεί τη διασύνδεση τριών επιμέρους κομματιών: τα δεδομένα από την μοντελοποίηση του κτιρίου στο Open Studio, τα δεδομένα από τους αισθητήρες και την επικοινωνία με το Gorgias Cloud για την λήψη των αποφάσεων. Το frontend έχει υλοποιηθεί με HTML και περιλαμβάνει το σύνολο των στοιχείων που παρουσιάζονται στον χρήστη στο Browser. Το backend της εφαρμογής είναι ο local server ο οποίος λειτουργεί με το Django.

Αρχικά ο χρήστης, ανοίγοντας την εφαρμογή, γίνεται σύνδεση στον local host:8080 ο οποίος κάνει render της αρχικής σελίδας (home). Μέσω αυτής της σελίδας επιλέγονται οι διαφορετικές τιμές και παράμετροι. Στην συνέχεια ελέγχεται αν στο πεδίο επιλογής χώρου είναι το “room2”, το οποίο είναι και το δωμάτιο που είναι συνδεδεμένο με τους αισθητήρες. Αν λοιπόν έχει επιλεγεί αυτός ο χώρος τότε μέσω του REST API γίνεται επικοινωνία με τον δίαυλο BAOS 777, ο οποίος επιστρέφει στον browser τις τιμές από τους αισθητήρες που διαθέτουμε. Έπειτα, το σύνολο των στοιχείων που έχει εισάγει ο χρήστης, περνάνε στο backend της εφαρμογής από όπου στέλνονται, μέσω REST API, στο Gorgias Cloud. Αρχικά γίνεται μέσω της εντολής “asserta” η εκχώρηση όλων των ερωτημάτων και στην συνέχεια με ένα δεύτερο αίτημα του server προς το Gorgias cloud, γίνεται η επαλήθευση (prove) των ερωτημάτων (queries). Η απάντηση που επιστρέφει περνάει μέσω του server και εμφανίζεται στον browser, όπου ο χρήστης μπορεί να δει τα αποτελέσματα. Θα πρέπει να σημειώσουμε ότι τα δεδομένα από το Open Studio έχουν ενοποιηθεί μέσα στο σύνολο των αρχείων που τρέχουν στο Gorgias Cloud.



ΕΙΚΟΝΑ 7. Διάγραμμα Αρχιτεκτονικής της εφαρμογής

## 4.2 REST API Gorgias Cloud

Η επικοινωνία μεταξύ της εφαρμογής που αναπτύχθηκε και του Gorgias Cloud, δεν θα ήταν εφικτή χωρίς την χρήση ενός REST API. Η βιβλιοθήκη για συγκεκριμένο REST API αναπτύχθηκε μέσω του Swagger για την γλώσσα προγραμματισμού Python. Η βασική δομή αυτού του προγράμματος είναι η εξής:

- Η σύνδεση στον Gorgias Cloud
- Η αποστολή ενός συνόλου ερωτημάτων (queries) προς το cloud
- Η επαλήθευση ή μη των προηγούμενων ερωτημάτων (prove)
- Η ανάκληση των ερωτημάτων (retract) από το cloud

Αναλυτικά μπορούμε των κώδικα που αναπτύχθηκε για την λειτουργία των παραπάνω.

Στην αρχή του κώδικα γίνεται η σύνδεση στο Gorgias cloud μέσω του configuration.

```
configuration = swagger_client.Configuration()
configuration.username = '****'
configuration.password = '****'
configuration.host = http://aiasvm1.amcl.tuc.gr:8085
```

Έπειτα γίνεται εύρεση του επιθυμητού αρχείου μέσα στον φάκελο στον οποίο βρίσκεται.

```
project = 'final'
api_response = api_instance2.consult_file_using_post('combinedPolicy.pl', project)
print(api_response)
```

Προκειμένου να διευκολύνουμε την εμφάνιση των αποτελεσμάτων ανά δωμάτιο και ανά συσκευή που υπάρχει μέσα σε αυτό, δημιουργούμε πέντε πίνακες. Ο κάθε ένας από αυτούς τους πίνακες περιέχει το σύνολο των επιλογών που είναι διαθέσιμες στην παρούσα εφαρμογή.

```
roomAssignment = {
    "room1": "space_121",
    "room2": "space_110",
    "room3": "space_115",
}

coolerAssignment = {
    "room1": "cooler1",
    "room2": "cooler2",
    "room3": "cooler3",
}

heaterAssignment = {
    "room1": "heater1",
    "room2": "heater2",
```

```

    "room3": "heater3",
}

alarmAssignment = {
    "room1": "alarm1",
    "room2": "alarm2",
    "room3": "alarm3",
}

sensorAssignment = {
    "room1": "sensor1",
    "room2": "sensor2",
    "room3": "sensor3",
}

```

Στην συνέχεια, χρησιμοποιείται η εντολή “asserta” για την αποστολή του συνόλου των πιθανών ερωτημάτων του χρήστη προς το Gorgias Cloud. Η μορφή κάθε αιτήματος είναι η εξής:

Στην γραμμή no.1 χρησιμοποιείται συνάρτηση “hasTemp” η οποία δέχεται δύο παραμέτρους σαν είσοδο. Η πρώτη μεταβλητή εισάγεται μέσω του πίνακα “roomAssignment” όπως φαίνεται στην γραμμή no.2. Η δεύτερη μεταβλητή είναι τύπου float καθώς περιέχει την τιμή της θερμοκρασίας που δίνει ο χρήστης από το frontend της εφαρμογής. Στην γραμμή no.5, no.6 δηλώνεται το πλήθος των απαντήσεων που επιθυμούμε να εμφανιστούν και ο χρόνος απόκρισης από τον Server, αντίστοιχα. Στην τελευταία γραμμή no.6, δηλώνεται η απάντηση που λάβαμε σε μια μεταβλητή ώστε να την εμφανίσουμε. Το σύνολο των εντολών παρατίθενται παρακάτω:

```

1 A1 = {"query": "asserta(hasTemp({},{}))."
2   format(roomAssignment[data["rooms"]],
3   int(float(data["currentTemp"]))),
4   "result_size": 1,
5   "time": 1000}
6 api_response = api_instance2.prolog_command_using_post(A1)

```

Αντίστοιχη διαδικασία γίνεται για όλα τα ερωτήματα (queries) όπως φαίνεται αναλυτικά στις γραμμές no.208-246 του κώδικα που βρίσκεται στο Appendix.

Στο επόμενο τμήμα του κώδικα γίνεται επαλήθευση των απαντήσεων με την εντολή “prove” και η εκτύπωση αυτών. Καθώς έχουμε δύο ειδών συσκευές (ψύξη-θέρμανση), γίνεται αρχικά ένας έλεγχος και, ανάλογα με το ποια συσκευή είναι ενεργοποιημένη, εκτελείται και το αντίστοιχο μέρος του κώδικα. Η δήλωση των εντολών γίνεται με αντίστοιχο τρόπο με προηγουμένως.

Στην γραμμή no.1 χρησιμοποιείται η συνάρτηση “switchOn” η οποία δέχεται σαν είσοδο δύο τιμές, το δωμάτιο και την συσκευή που υπάρχει σε αυτό. Την πρώτη τιμή, όπως βλέπουμε στην γραμμή no.2, την δέχεται μέσω του πίνακα “roomAssignment”. Η δεύτερη τιμή εισάγεται μέσω του πίνακα “coolerAssignmen”, όπως φαίνεται στην γραμμή no.3. Στην γραμμή no.5, no.6 δηλώνεται το πλήθος των απαντήσεων που επιθυμούμε να εμφανιστούν και ο χρόνος απόκρισης από τον Server,

αντίστοιχα. Στην γραμμή no.6, δηλώνεται η απάντηση που λάβαμε σε μια μεταβλητή. Τέλος στην γραμμή no.7 εμφανίζεται στον χρήστη μέσω του UI η επιθυμητή απάντηση. Το σύνολο των εντολών παρατίθενται παρακάτω:

```
1 P1 = {"query": "prove([switchOn({},{}),E)"
2 format(roomAssignment[data['rooms']],
3 coolerAssignment[data['rooms']]),
4 "result_size": 1,
5 "time": 10000}
6 api_response = api_instance2.prove_using_post(P1)
7 result += 'Switch On cooler: ' + makeResult(api_response)
```

Αντίστοιχη διαδικασία γίνεται για όλα τα ερωτήματα (queries) όπως φαίνεται αναλυτικά στις γραμμές no.248-317 του κώδικα που βρίσκεται στο Appendix.

Στο τελευταίο κομμάτι του REST API γίνεται χρήση της εντολής “retract”. Αυτό το κομμάτι του κώδικα επιτελεί μια σημαντική λειτουργία η οποία είναι ο καθαρισμός του Gorgias Cloud από το σύνολο των queries που έχουν γίνει στα προηγούμενα βήματα. Αυτό συμβαίνει προκειμένου τα δεδομένα που χρησιμοποιήθηκαν σε ένα παράδειγμα να αφαιρεθούν από την μνήμη της εφαρμογής ώστε να μην επηρεάσουν τα αποτελέσματα μελλοντικών χρήσεων.

Στην γραμμή no.1 χρησιμοποιείται η συνάρτηση “hasTemp” η οποία δέχεται σαν είσοδο δύο μεταβλητές που περιέχουν το σύνολο των δεδομένων που έχουν εισαχθεί κατά την εκτέλεση του προγράμματος. Στην γραμμή no.2 δηλώνεται το πλήθος των απαντήσεων που επιθυμούμε να εμφανιστούν. Σε αυτή την περίπτωση, σε αντίθεση με τα προηγούμενα παραδείγματα, βάζουμε σαν τιμή το “20” που είναι μια μεγάλη τιμή ικανή να κάνει retract το σύνολο των δεδομένων που έχουν εκχωρηθεί. Στην γραμμή no.3 δηλώνεται ο χρόνος απόκρισης από τον Server. Στην τελευταία γραμμή no.4, δηλώνεται η απάντηση που λάβαμε σε μια μεταβλητή ώστε να την εμφανίσουμε.

```
1 R1 = {"query": "retract(hasTemp(X,Y)).",
2 "result_size": 20,
3 "time": 1000}
4 retract = api_instance2.prolog_command_using_post(R1)
```

Αντίστοιχη διαδικασία γίνεται για όλα τα ερωτήματα (queries) όπως φαίνεται αναλυτικά στις γραμμές no.319-342 του κώδικα που βρίσκεται στο Appendix.

### 4.3 REST API BAOS 777

Προκειμένου να γίνει διαχείριση των δεδομένων που λαμβάνουμε από το BAOS 777 και η ενσωμάτωση τους στο σύστημα λήψης αποφάσεων, δημιουργήθηκε ένα REST API. Το συγκεκριμένο περιβάλλον διεπαφής δημιουργήθηκε σε Python χρησιμοποιώντας την βιβλιοθήκη που παρέχει ο κατασκευαστής [16].

Ο κώδικας ο οποίος αναπτύχθηκε παρατίθεται στην συνέχεια:

Αρχικά γίνεται η εισαγωγή όλων των απαραίτητων βιβλιοθηκών.

```
import requests
import json
from collections import namedtuple
import sched, time
import os
import os.path
import csv
from datetime import datetime
clear = lambda: os.system('cls')
clear()
```

Στην συνέχεια ανοίγουμε το κανάλι επικοινωνίας με την συσκευή, για την αποστολή των αιτημάτων.

```
s = requests.Session()
```

Χρησιμοποιούμε σαν header τις τιμές (values) βάσει των προδιαγραφών που διαθέτουμε για την σωστή επικοινωνία.

```
headers = {'Content-type': 'application/json, text/plain, /*'}
s.headers.update(headers);
s.headers.update({
    'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36', })
headers = {'Accept-Language': 'el-GR,el;q=0.9,en;q=0.8'}
s.headers.update(headers);
```

Γίνεται σύνδεση στην συσκευή.

```
url = 'http://147.27.33.130/rest/login'
pload = {'username': '****', 'password': '****'}
```

Έπειτα διαβάζουμε το device status το οποίο μας επιστρέφει σαν τιμή, το όνομα της συσκευής καθώς και την έκδοση της (version).

```
URL = 'http://147.27.33.130/rest/device'
r = s.get(URL)
x = json.loads(r.text, object_hook=lambda d: namedtuple('X',
d.keys())(*d.values()))
DevName = "Device {}, Version {}".format(x.device.name,
x.device.build_version)
Devfilename = "{}_{}.csv".format(x.device.name, x.device.build_version)
```



Σε αυτό το σημείο δημιουργούμε την συνάρτηση “do\_something” η οποία καλείται για να διαβάσει τις τιμές από τις συσκευές που είναι συνδεδεμένες στο BAOS 777.

```
def do_something(sc):
```

```
    clear()
```

Διαβάζονται οι τιμές από το BAOS 777.

```
    URL = 'http://147.27.33.130/rest/datapoints/values?start=1&end=25'
    r = s.get(URL)
    x = json.loads(r.text, object_hook = lambda d : namedtuple('X', d.keys())(*d.values()))
    print(DevName)
    print('-----')
    for i in range(DevCount):
        txt3 = "{:<12} Id {} {} is_update {} is_valid {} value {} time
{}".format(DevNames.datapoints_descriptions[i].name, x.datapoints_values[i].id,
x.datapoints_values[i].Format, x.datapoints_values[i].state.is_update,
x.datapoints_values[i].state.is_valid, x.datapoints_values[i].value, datetime.now())
        print(txt3)
```

Γίνεται format των δεδομένων με τα επιθυμητό τρόπο ώστε στην συνέχεια να αποθηκευτούν σε ένα αρχείο csv.

```
        txt3 = "{0,0,0,0,0,0}".format(DevNames.datapoints_descriptions[i].name,
x.datapoints_values[i].id, x.datapoints_values[i].Format, x.datapoints_values[i].state.is_update,
x.datapoints_values[i].state.is_valid, x.datapoints_values[i].value )
```

Τέλος, γράφει όλα τα δεδομένα σε ένα αρχείο csv ώστε να τα διαβάσουμε στην συνέχεια της εφαρμογής.

```
with open(Devfilename, 'a+', newline='') as file:
    writer = csv.writer(file)
    writer.writerow([txt3])
```

## 4.4 Εφαρμογή του Django

Προκειμένου να δημιουργηθεί μια ολοκληρωμένη εφαρμογή με αυξημένες δυνατότητες, χρησιμοποιήθηκε το πλαίσιο ανάπτυξης εφαρμογών Django. Μέσω αυτού που πλαισίου κατέστη δυνατό να δημιουργηθεί, με τα εργαλεία που διατίθενται, τόσο το frontend όσο και το backend της εφαρμογής μας. Το frontend είναι το μέρος που τρέχει το πρόγραμμα περιήγησης (browser) και αποτελεί το σημείο διεπαφής του χρήστη με την εφαρμογή. Στο backend βρίσκεται αποθηκευμένο το πρόγραμμα και η βάση δεδομένων τα οποία καλούνται μέσω του browser.

Η Django δημιουργείται απευθείας μέσω του Pyscharm και αυτόματα δημιουργείται και ένας φάκελος με όλα τα απαραίτητα αρχεία για την υλοποίηση της εφαρμογής μας. Τα αρχεία που δημιουργούνται ένα τα εξής:

- **Urls.py**

Πρόκειται για έναν πίνακα ο οποίος συνδέει τις διαφορετικές διευθύνσεις (url) που είναι απαραίτητες στην εφαρμογή μας με συναρτήσεις. Η πρώτη συνάρτηση “views.index”, αφορά το frontend και το σύνολο των στοιχείων που επιθυμούμε να εμφανίζονται όταν ανοίγει η εφαρμογή. Η δεύτερη συνάρτηση “view.gorgias” αφορά την επικοινωνία της εφαρμογής μας με το gorgias cloud όπου είναι το σημείο που λαμβάνονται οι αποφάσεις. Η τρίτη συνάρτηση “view.knx\_baos\_csv” αναφέρεται στην επικοινωνία με τον δίαυλο επικοινωνίας KNX BAOS 777 και χρησιμοποιείται όποτε χρειαζόμαστε τα δεδομένα από τους αισθητήρες.

- **Views.py**

Αποτελεί το κύριο αρχείο μέσα στο οποίο βρίσκεται το μεγαλύτερο μέρος του κώδικα που εκτελείται. Αποτελείται από δύο βασικές συναρτήσεις: “knx\_baos\_csv”, “gorgias”. Η πρώτη συνάρτηση αναφέρεται στο REST Api που έχει δημιουργηθεί με σκοπό την λήψη των δεδομένων των αισθητήρων μέσω του δίαυλου BAOS 777. Η δεύτερη συνάρτηση αποτελεί το REST Api του Gorgias Cloud, που σκοπό έχει την μεταφορά όλων των ερωτημάτων (queries) που γίνονται μέσω του frontend και την λήψη της απάντησης μετά από επεξεργασία. Αυτές οι δύο συναρτήσεις λειτουργούν συνδυαστικά ώστε να επιτευχθεί ένα άρτιο αποτέλεσμα.

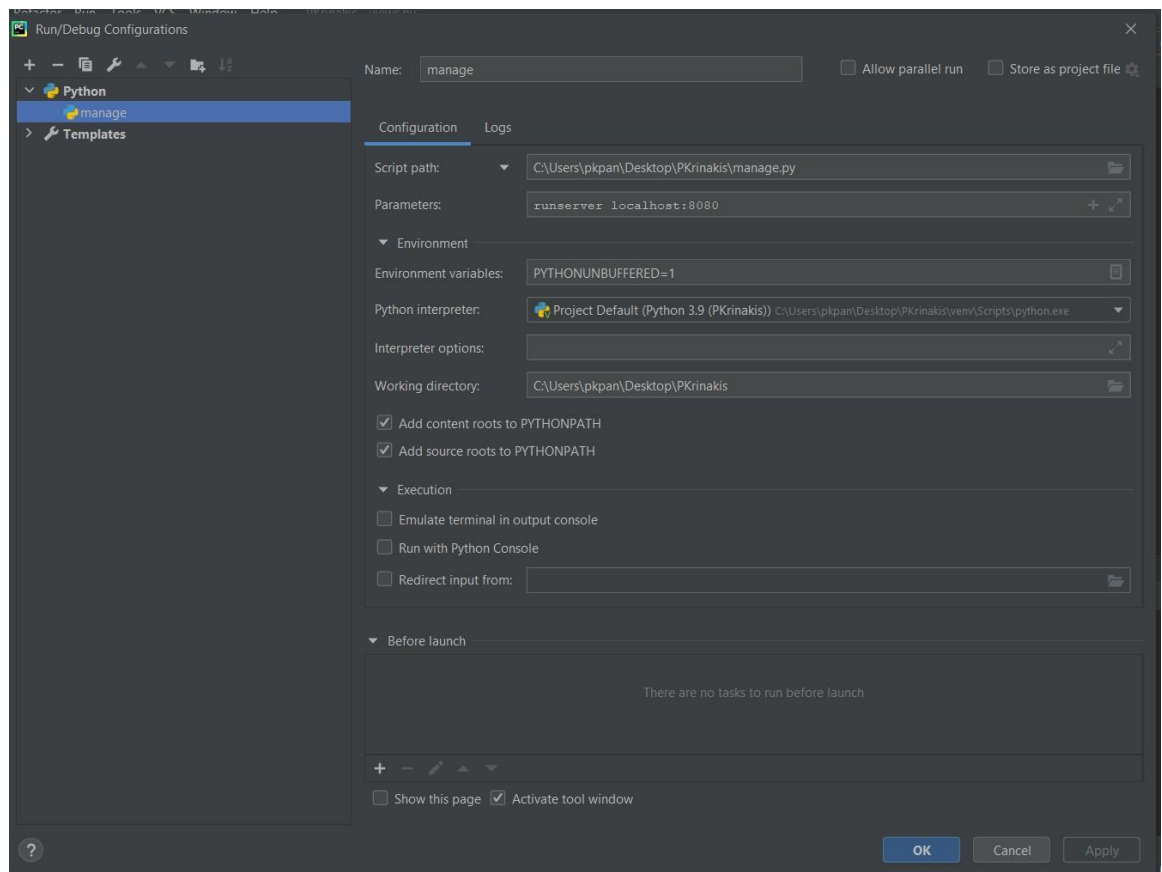
- **Settings.py**

Το αρχείο αυτό παράγεται αυτόματα κατά τη δημιουργία της Django εφαρμογής. Μέσα σε αυτό εμπεριέχονται όλες οι βασικές ρυθμίσεις που χρησιμοποιούνται. Αυτές οι ρυθμίσεις είναι προεπιλεγμένες και δεν χρειάστηκαν να αλλαχθούν στην παρούσα εφαρμογή.

- **Manage.py**

Πρόκειται για ένα ακόμη αρχείο που παράγεται αυτόματα κατά τη δημιουργία της Django εφαρμογής. Μέσω αυτού του αρχείου παραμετροποιούνται οι προεπιλεγμένες ρυθμίσεις που χρειάζεται η εφαρμογή. Τέλος είναι το αρχείο που χρησιμοποιείται για την εκτέλεση της εφαρμογής. Στην ΕΙΚΟΝΑ 4. Φαίνονται και οι ρυθμίσεις που χρησιμοποιούνται για την εκτέλεση της εφαρμογής μέσω του αρχείου manage.py. Συγκεκριμένα μπορούμε να

δούμε σαν παράμετρο την διεύθυνση του τοπικού διακομιστή (local server) που έχει δημιουργηθεί και πάνω στον οποίο εκτελείται η εφαρμογή.



ΕΙΚΟΝΑ 8. Ρυθμίσεις εκτέλεσης του Django

## 4.5 Το γραφικό περιβάλλον του συστήματος

Το γραφικό περιβάλλον του συστήματος, που αποτελεί το frontend της εφαρμογής, δημιουργήθηκε μέσω HTML, JavaScript αλλά και CSS. Το αρχείο home.html περιέχει όλο τον κώδικα που χρησιμοποιήθηκε για την δημιουργία του frontend. Αναλυτικά:

Η CSS χρησιμοποιείται για την βασική μορφοποίηση της σελίδας και την βελτίωση της εμφάνισης αυτής, όπως φαίνεται στο κώδικα παρακάτω:

```
<style>
body {
```

```
padding: 15px;
}

form div, button {
margin: 10px;
}
form label {
font-size: large;
font-weight: bold;
margin-right: 5px;
}
```

Η HTML αποτελείται από ένα σύνολο ετικετών (tags) που ελέγχουν την δομή του κειμένου που εμφανίζεται στην σελίδα. Επιπλέον παρέχουν πληροφορίες προς το πρόγραμμα περιήγησης σχετικές με την σελίδα που πρόκειται να εμφανίσουν, όπως τον τίτλο. Στην συγκεκριμένη σελίδα έχουν χρησιμοποιηθεί διαφορετικές ετικέτες ανάλογα με το είδος των στοιχείων που θέλουμε να εισάγει ο χρήστης κάθε φορά. Συγκεκριμένα, παρατηρώντας την ΕΙΚΟΝΑ 5, θα αναλυθούν οι επιλογές από πάνω προς το κάτω μέρος:

Η πρώτη επιλογή του χρήστη αναφέρεται στο δωμάτιο (room) που επιθυμεί να γίνει χρήση της εφαρμογής. Για την επιλογή αυτή χρησιμοποιείται ένα drop down menu που περιέχει τρεις διαφορετικές επιλογές δωματίων, Auditorium, Office(sensor) και Computer room.

```
<div>
<label for="rooms">Rooms</label>
<select name="rooms" id="rooms" onchange="onSelectChange(this)">
<option value="room1">Auditorium</option>
<option value="room2">Office (Sensor)</option>
<option value="room3">Computer room</option>
</select>
</div>
```

Αντίστοιχα λειτουργεί και το δεύτερο menu επιλογών, μέσω drop down menu. Σε αυτό το πεδίο ο χρήστης επιλέγει την λειτουργία (action), δηλαδή αν επιθυμεί να χρησιμοποιήσει την ψύξη (cool) ή την θέρμανση (heat) για το χώρο που έχει επιλέξει.

```
<div>
<label for="action">Action</label>
<select name="action" id="action">
<option value="cool">Cool</option>
<option value="heat">Heat</option>
</select>
</div>
```

Οι επόμενες τρεις επιλογές αφορούν την κατάσταση των συσκευών που υπάρχουν στο χώρο, δηλαδή αν οι συσκευές είναι ενεργοποιημένες (On) ή απενεργοποιημένες (Off). Οι συσκευές που

υπάρχουν είναι το σύστημα ψύξης (cooler), το σύστημα θέρμανσης (heater) καθώς και ο συναγερμός (alarm). Ο τρόπος που εμφανίζονται στην οθόνη είναι μέσω checkboxes που μπορεί ο χρήστης να επιλέξει ή αποεπιλέξει αν θέλει να είναι ενεργοποιημένο ή απενεργοποιημένο το σύστημα, αντίστοιχα.

```
<div>  
  <label for="coolerState">Cooler</label>  
  <input id="coolerState" type="checkbox">  
</div>
```

```
<div>  
  <label for="heaterState">Heater</label>  
  <input id="heaterState" type="checkbox">  
</div>
```

```
<div>  
  <label for="alarmState">Alarm</label>  
  <input id="alarmState" type="checkbox">  
</div>
```

Οι επόμενες δύο επιλογές αναφέρονται στην υπάρχουσα θερμοκρασία (current temperature) και στην θερμοκρασία στόχος (target temperature). Σε αυτά τα πεδία εισάγονται νούμερα ανάλογα με τις τιμές που επιθυμεί ο χρήστης.

```
<div>  
  <label for="currentTemp">Current Temperature</label>  
  <input id="currentTemp" type="number">  
</div>
```

```
<div>  
  <label for="targetTemp">Target Temperature</label>  
  <input id="targetTemp" type="number">  
</div>
```

Έπειτα από τον ορισμό θερμοκρασίας υπάρχει ένα ακόμη checkbox το οποίο αναφέρεται στον αισθητήρα κίνησης (motion). Μέσω αυτού του πεδίου ο χρήστης μπορεί να ορίσει αν υπάρχει ή όχι κίνηση στον χώρο.

```
<div>  
  <label for="motionState">Motion</label>  
  <input id="motionState" type="checkbox">  
</div>
```

Όμοιο με τα πεδία της θερμοκρασία είναι και το πεδίο στο οποίο εισάγεται η τιμή του διοξειδίου του άνθρακα (Current CO<sub>2</sub>).

```

<div>
  <label for="currentCO2">Current CO2</label>
  <input id="currentCO2" type="number">
</div>

```

Στο πεδίο της ώρας (current time), εισάγεται η ώρα σε μορφή δωδεκάωρου, χωρισμένη σε προ (am) και μετά μεσημβρίας (pm).

```

<div>
  <label for="currentTime">Current Time</label>
  <input id="currentTime" type="time">
</div>

```

Τέλος τα δύο τελευταία πεδία αναφέρονται στην ενεργειακή πολιτική που μπορεί να επιλέξει ο χρήστης. Συγκεκριμένα στο πρώτο πεδίο (energy policy) υπάρχει μια drop down list η οποία περιέχει τρεις επιλογές: none, saving, critical. Στο δεύτερο πεδίο (Thermal zone level) υπάρχει επίσης μια drop down list, η οποία περιέχει τιμές από το ένα (1) μέχρι το δώδεκα (12), ανάλογα με το επίπεδο ενεργειακής εξοικονόμησης που θέλει να πετύχει ο χρήστης.

```

<div>
  <label for="energyPolicy">Energy policy</label>
  <select name="energyPolicy" id="energyPolicy">
    <option value="none">None</option>
    <option value="saving">Saving</option>
    <option value="critical">Critical</option>
  </select>
</div>

```

```

<div>
  <label for="thermalZoneLevel">Thermal zone level</label>
  <select name="thermalZoneLevel" id="thermalZoneLevel">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
    <option value="6">6</option>
    <option value="7">7</option>
    <option value="8">8</option>
    <option value="9">9</option>
    <option value="10">10</option>
    <option value="11">11</option>
    <option value="12">12</option>
  </select>
</div>

```

Στο κάτω μέρος της σελίδας υπάρχει το κουμπί “prove” το οποίο με πάτημα του τρέχει την αντίστοιχη συνάρτηση με σκοπό την μεταφορά των δεδομένων από το frontend στο backend και την επεξεργασία αυτών.

```
button type="button" id="proveBtn" onclick="prove()">Prove</button>
```

## Automated decision making tool

**Rooms**

**Action**

**Cooler** ☐

**Heater** ☐

**Alarm** ☐

**Current Temperature**

**Target Temperature**

**Motion** ☐

**Current CO2**

**Current Time**

**Energy policy**

**Thermal zone level**

ΕΙΚΟΝΑ 9. Βασικό Μενού επιλογών του εργαλείου

Η JavaScript έχει τον ρόλο της επικοινωνίας μεταξύ των πεδίων που έχουν δημιουργηθεί με την HTML και του backend με στόχο της λήψης της απόφασης. Αρχικά στο πρώτο μέρος του κώδικα βρίσκουμε όλα τα UI elements, δηλαδή τα inputs που έχουν εισαχθεί μέσω της HTML, και καταχωρούνται ως μεταβλητές ώστε να μπορούν να διαβαστούν από το πρόγραμμα.

```
const rooms = document.getElementById("rooms");  
const action = document.getElementById("action");  
const coolerState = document.getElementById("coolerState");  
const heaterState = document.getElementById("heaterState");  
const alarmState = document.getElementById("alarmState");
```

```

const currentTemp = document.getElementById("currentTemp");
const targetTemp = document.getElementById("targetTemp");
const motionState = document.getElementById("motionState");
const currentCO2 = document.getElementById("currentCO2");
const currentTime = document.getElementById("currentTime");
const energyPolicy = document.getElementById("energyPolicy");
const thermalZoneLevel = document.getElementById("thermalZoneLevel");
const proveBtn = document.getElementById("proveBtn");
const resultsHeader = document.getElementById("resultsHeader");
const resultsParagraph = document.getElementById("resultsParagraph");

```

Στην συνέχεια δημιουργούμε την συνάρτηση “clearInputs” η οποία καθαρίζει την σελίδα από τιμές που μπορεί να υπάρχουν, εισάγοντας στις μεταβλητές τιμές ανάλογα το είδος τους.

```

function clearInputs (data) {
  action.value = 'cool';
  coolerState.checked = false;
  heaterState.checked = false;
  alarmState.checked = false;
  currentTemp.value = null;
  targetTemp.value = null;
  motionState.checked = false;
  currentCO2.value = null;
  currentTime.value = null;
  energyPolicy.value = 'none';
  thermalZoneLevel.value = '1';
  resultsHeader.hidden = true;
  resultsParagraph.textContent = "";
}

```

Η επόμενη συνάρτηση είναι η “fetchRoomState” η οποία χρησιμοποιείται για λήψη των τιμών από τους αισθητήρες που διαθέτουμε μέσω του KNX BAOS 777. Ουσιαστικά με αυτό τον τρόπο συμπληρώνονται αυτόματα οι τιμές στα πεδία: Current Temperature, Motion, Current CO<sub>2</sub>, Current Time.

```

async function fetchRoomState () {
  let response = await fetch('/api/knx_baos_csv');
  let data = await response.json();
  console.log(data);
  currentTemp.value = data['T'];
  currentCO2.value = data['CO2'];
  motionState.checked = data['illuminance'] >= 150;
  const now = new Date();
  currentTime.value = `${now.getHours()}:${now.getMinutes()}`;
}

```

Συμπληρωματικά λειτουργεί και η συνάρτηση “onSelectChange” η οποία ενεργοποιείται και καλώντας την συνάρτηση “fetchRoomState” μόνο για την περίπτωση που ο χρήστης επιλέξει ως δωμάτιο την δεύτερη επιλογή, room2, που είναι και χώρος όπου έχουμε ορίσει ότι είναι τοποθετημένοι οι αισθητήρες.



```
function onSelectChange (el) {
  console.log('changed!', el.value);
  clearInputs();
  if (el.value === 'room2') {
    fetchRoomState();
  }
}
```

Η τελευταία συνάρτηση που έχει υλοποιηθεί είναι η “prove”. Η συνάρτηση αυτή ενεργοποιείται όταν ο χρήστης πατήσει το κουμπί “prove”, το οποίο υπάρχει στο τέλος της σελίδας. Με αυτό τον τρόπο όλα τα πεδία που έχουν εισαχθεί από τον χρήστη καταχωρούνται σε μεταβλητές τις οποίες εκχωρεί σε έναν τύπο δεδομένων .json ώστε να σταλεί το αίτημα (request) στον Gorgia. Ταυτόχρονα ενεργοποιείται η συνάρτηση “gorgias” η οποία βρίσκεται στο backend μέρος του κώδικα της Django, στο αρχείο view.py. Τα δεδομένα τα οποία λαμβάνει ως αποτελέσματα της κλήσης της συνάρτησης εμφανίζονται στο κάτω μέρος την σελίδας.

```
async function prove () {
  resultsHeader.hidden = true;
  resultsParagraph.textContent = "";

  const data = {
    rooms: rooms.value,
    action: action.value,
    coolerState: coolerState.checked,
    heaterState: heaterState.checked,
    alarmState: alarmState.checked,
    currentTemp: currentTemp.value,
    targetTemp: targetTemp.value,
    motionState: motionState.checked,
    currentCO2: currentCO2.value,
    currentTime: currentTime.value,
    energyPolicy: energyPolicy.value,
    thermalZoneLevel: thermalZoneLevel.value,
  }

  let response = await fetch('/api/gorgias', {
    method: 'POST',
    mode: 'cors', // no-cors, *cors, same-origin
    cache: 'no-cache', // *default, no-cache, reload, force-cache, only-if-cached
    credentials: 'same-origin', // include, *same-origin, omit
    headers: {
      'Content-Type': 'application/json'
    },
    redirect: 'follow', // manual, *follow, error
    referrerPolicy: 'no-referrer', // no-referrer, *no-referrer-when-downgrade, origin, origin-when-cross-origin, same-origin, strict-origin, strict-origin-when-cross-origin, unsafe-url
    body: JSON.stringify(data) // body data type must match "Content-Type" header
  });
  let result = await response.json();
  console.log(result);
  resultsHeader.hidden = false;
}
```

```
resultsParagraph.textContent = result;  
}
```

## 4.6 Παραμετροποίηση επιλογών στο Open Studio

Παρακάτω αναλύονται οι βασικές λειτουργίες του προγράμματος καθώς και οι επιλογές που χρησιμοποιήθηκαν κατά την μοντελοποίηση του Κτιρίου K2 της Σχολής Μηχανικών Περιβάλλοντος.

- **Site (Τοποθεσία)**

Αποτελεί την πρώτη καρτέλα επιλογών του προγράμματος στην οποία εισάγεται το αρχείο καιρού (weather file). Στην παρούσα εργασία χρησιμοποιήθηκε το αρχείο Soudhour.erw το οποίο περιέχει τα δεδομένα καιρού για την περιοχή του Ακρωτηρίου στα Χανιά στην οποία και βρίσκεται το κτίριο. Συνολικά περιέχονται 30 διαφορετικές μετρήσεις που έχουν ληφθεί ανά μια ώρα στην διάρκεια ενός ημερολογιακού έτους. Μεταξύ αυτών οι σημαντικότερες είναι: θερμοκρασία, υγρασία, ατμοσφαιρική πίεση, ταχύτητα και διεύθυνση ανέμου, αλλά και ηλιοφάνεια.

- **Schedules (Χρονοδιαγράμματα)**

Σε αυτή την καρτέλα δημιουργούνται τα χρονοδιαγράμματα βάσει των οποίων γίνεται η προσομοίωση. Τα χρονοδιαγράμματα εφαρμόζονται «σέρνοντας» από τη δεξιά πλευρά της οθόνης, My Model, την επιλογή που μας ενδιαφέρει, τοποθετώντας την στο αντίστοιχο κελί. Συγκεκριμένα, στο παρόν πλάνο, χρησιμοποιήθηκε το κύριο ωράριο λειτουργίας του κτιρίου που είναι 8:00 με 16:00. Επιπλέον αναφορικά με την ψύξη και την θέρμανση δημιουργήθηκε ένα πλάνο συνδυάζοντας τις αντίστοιχες περιόδους αλλά και θέτοντας άνω και κάτω όριο, 19C° και 26C° αντίστοιχα.

- **Constructions (Κατασκευές)**

Πρόκειται για την καρτέλα στην οποία εισάγονται τα υλικά από τα οποία αποτελούνται όλες οι διαφορετικές επιφάνειες του κελύφους. Συγκεκριμένα για το παρόν κτίριο δημιουργήθηκαν σε διαφορετικά construction sets, ένα για το ισόγειο, ένα για τον 1° όροφο και ένα για 2° όροφο. Το κάθε ένα από τα σετ κατασκευών αποτελείται από επιμέρους στοιχεία όπως οι εσωτερικοί-εξωτερικοί τοίχοι, το πάτωμα, οι πόρτες και τα παράθυρα. Συνεπώς δημιουργούνται κατασκευαστικά κομμάτια από διαφορετικά υλικά με διαφορετικές ιδιότητες ανάλογα με αποτέλεσμα και τα χαρακτηριστικά που θέλουμε να έχει ένας χώρος, ένας όροφος ή ακόμη και ολόκληρο το κτίριο.

- **Loads (Φορτία)**

Σε αυτή την καρτέλα ορίζουμε το σύνολο των φορτίων τα οποία υπάρχουν στον κτίριο. Αυτά τα φορτία στην συνέχεια μπορούν να αντιστοιχιστούν στους διαφορετικούς χώρους που έχουν δημιουργήσει. Οι κατηγορίες των φορτίων που έχουν χρησιμοποιήσει είναι οι εξής:

- Πλήθος ατόμων
- Ενέργεια από λαμπτήρες
- Ενέργεια από ηλεκτρικό εξοπλισμό

- **Space Types (Είδη χώρων)**

Σε αυτό το σημείο μπορούμε να δημιουργήσουμε τα διαφορετικά είδη των χώρων που επιθυμούμε καθώς επίσης και να εφαρμόσουμε τα σετ κατασκευών και τα χρονοδιαγράμματα ανάλογα με την χρήση του κάθε χώρου. Το παρόν κτίριο χωρίστηκε σύμφωνα με τις ανάγκες του σε 4 κατηγορίες:

- Εργαστήρια
- Γραφεία
- Κοινόχρηστοι χώροι
- Σοφίτα

- **Facility (Εγκατάσταση)**

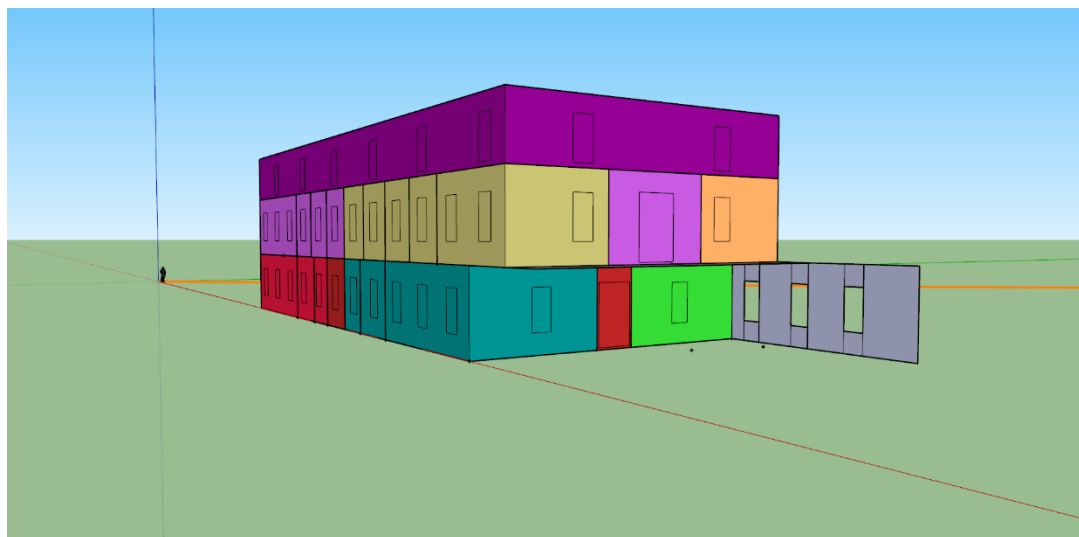
Σε αυτή την καρτέλα μπορεί να δούμε μια προεπισκόπηση του κτιρίου ανά όροφο, χρήση χώρου ή θερμική ζώνη επιλέγοντας αντίστοιχα μέσω από το menu το πεδίο Render By. Επιπλέον μπορούμε να ελέγξουμε τις παραμέτρους του μοντέλου αλλά και να κάνουμε τροποποιήσεις καθώς μας δίνει μια ιεραρχικά κατανεμημένη εικόνα του μοντελοποιημένου κτηρίου.

- **Spaces (Χώροι)**

Το κτίριο έχει χωριστεί σε συνολικά 30 διαφορετικούς χώρους για την κάλυψη του συνόλου των αναγκών που υπάρχουν. Συνεπώς σε αυτή την καρτέλα μπορούμε να δούμε την αντιστοίχιση κάθε ενός από αυτούς τους χώρους στις θερμικές ζώνες στις οποίες ανήκουν καθώς επίσης και να οριστεί ο τύπος του κάθε χώρου σύμφωνα με τα είδη που έχουμε θέσει (εργαστήρια, γραφεία, κοινόχρηστοι χώροι και η σοφίτα).

- **Thermal Zones (θερμικές ζώνες)**

Πρόκειται για μια από τις σημαντικότερες καρτέλες στο πρόγραμμα καθώς εδώ μπορούμε να δούμε και να ορίσουμε τα συστήματα ψύξης/θέρμανσης (HVAC systems) για κάθε θερμική ζώνη του κτιρίου. Αξίζει να σημειώσουμε ότι ο χωρισμός σε θερμικές ζώνες γίνεται σύμφωνα με 3 κριτήρια που είναι: ο προσανατολισμός, ο όροφος αλλά και οι ιδιότητες του χώρου. Σύμφωνα με αυτά τα κριτήρια ολόκληρο το κτίριο χωρίστηκε σε 13 διαφορετικές θερμικές ζώνες. Τέλος για τη θερμική ζώνη 30, η οποία είναι η σοφίτα, είναι η μόνη για την οποία δεν έχουν εφαρμοστεί τα συστήματα ψύξης/θέρμανσης καθώς δεν τα διαθέτει.



ΕΙΚΟΝΑ 10. Οι διαφορετικές θερμικές ζώνες του κτιρίου

- **Run Simulation (Εναρξη προσομοίωσης)**

Σε αυτή την τελευταία καρτέλα γίνεται ο έλεγχος του μοντέλου για σφάλματα και έπειτα “τρέχει” η διαδικασία της προσομοίωσης του κτιρίου. Αν δεν υπάρχουν σφάλματα και η διαδικασία ολοκληρωθεί με επιτυχία μπορούμε να δούμε τα αποτελέσματα πατώντας το Show Report.

#### 4.6.1 Αποτελέσματα από το Open Studio

Ο φάκελος “K2vheating\_cooling” που δημιουργείται έπειτα από την εφαρμογή του simulation. Περιλαμβάνει πλήθος δεδομένων τα οποία απαιτείται να αναλυθούν ώστε να χρησιμοποιηθούν

αποκλειστικά αυτά τα οποία μπορούν να συμβάλουν στην λήψη ορθότερων αποφάσεων. Συγκεκριμένα μέσα στο φάκελο περιέχεται μια συγκεντρωτική αναφορά με τα αποτελέσματα που προέκυψαν.

Η αναφορά αυτή είναι χωρισμένη σε 21 διαφορετικές κατηγορίες με λεπτομέρειες αναφορικά με το είδος του χώρου (space types), τις θερμικές ζώνες (thermal zones), το σύστημα ψύξης/θέρμανσης (HVAC) αλλά και τις καταναλώσεις αυτών.

Στον παρακάτω πίνακα μπορούμε να δούμε τις καταναλώσεις σε Giga Joule χωρισμένες στις εξής κατηγορίες: ψύξη, θέρμανση, φωτισμός και ηλεκτρικές συσκευές.

ΠΙΝΑΚΑΣ 1. Συνολικές καταναλώσεις ανά είδος

	Electricity [GJ]	Natural Gas [GJ]	Additional Fuel [GJ]	District Cooling [GJ]	District Heating [GJ]	Water [m3]
Heating	0.00	0.00	0.00	0.00	52.20	0.00
Cooling	0.00	0.00	0.00	254.69	0.00	0.00
Interior Lighting	21.97	0.00	0.00	0.00	0.00	0.00
Exterior Lighting	0.00	0.00	0.00	0.00	0.00	0.00
Interior Equipment	171.73	0.00	0.00	0.00	0.00	0.00
Exterior Equipment	0.00	0.00	0.00	0.00	0.00	0.00
Fans	0.00	0.00	0.00	0.00	0.00	0.00
Pumps	0.00	0.00	0.00	0.00	0.00	0.00
Heat Rejection	0.00	0.00	0.00	0.00	0.00	0.00
Humidification	0.00	0.00	0.00	0.00	0.00	0.00
Heat Recovery	0.00	0.00	0.00	0.00	0.00	0.00
Water Systems	0.00	0.00	0.00	0.00	0.00	0.00
Refrigeration	0.00	0.00	0.00	0.00	0.00	0.00
Generators	0.00	0.00	0.00	0.00	0.00	0.00
Total End Uses	193.70	0.00	0.00	254.69	52.20	0.00

Προκειμένου να μπορέσουμε να αξιοποιήσουμε τα δεδομένα που μας παρέχονται από την μοντελοποίηση του κτιρίου K2 στο Open Studio, έγινε μια επιλογή τεσσάρων πηγών κατανάλωση ενέργειας. Οι τέσσερες πηγές αυτές είναι: η θέρμανση (heating), η ψύξη (cooling), ο ηλεκτρονικός εξοπλισμός (equipment) και ο φωτισμός (lighting). Για κάθε μια από αυτές τις κατηγορίες, μέσω του Open Studio, εξάχθηκε ένα αρχείο .csv. Τα αρχεία αυτά περιέχουν τις καταναλώσεις ανά μια ώρα για την διάρκεια ενός ολόκληρου έτους, ανά θερμική ζώνη, εκφρασμένες σε Joule. Για να μπορέσουμε να χρησιμοποιήσουμε αυτά τα δεδομένα έπρεπε να τα επεξεργασθούμε με τον εξής τρόπο:

Το άθροισμα των καταναλώσεων ανά θερμική ζώνη σε Joule:

$$\sum_1^n Energy\ measurement = x\ Joule$$

Η μετατροπή του αποτελέσματος σε kWh:

$$1J = 2.777778 \cdot 10^{-7} kWh$$

Τέλος για μπορούν τα δεδομένα να είναι συγκρίσιμα, το άθροισμα από κάθε θερμική ζώνη ξεχωριστά διαιρέθηκε με τα τετραγωνικά που καλύπτει:

$$\frac{total\ kWh}{thermal\ zone\ m^2} = x\ kWh/m^2$$

Τα δεδομένα από τις τέσσερις αυτές κατηγορίες συγκεντρώθηκαν στον παρακάτω πίνακα. Αξίζει να σημειωθεί ότι στην κατηγορία του εξοπλισμού (equipment) ορισμένες θερμικές ζώνες παρουσιάζουν μηδενικές καταναλώσεις λόγω του ότι πρόκειται για κοινόχρηστους χώρους.

ΠΙΝΑΚΑΣ 2. Καταναλώσεις ανά θερμική ζώνη

	thermal zone 17	thermal zone 18	thermal zone 19	thermal zone 20	thermal zone 21	thermal zone 22
cooling	5.69E+03	10710.70234	1916.346382	11569.6714	7974.905943	910.5762766
heating	1.12E+03	4.91E+02	4.52E+02	6.08E+02	9.56E+02	7.17E+02
equipment	1.57E+03	9963.946591	0	9963.946591	9963.946591	0
lights	4.52E+02	941.9697407	37.67878963	941.9697407	904.2909511	37.67878963
total KWh	8829.20	22107.70	2406.01	23083.68	19798.78	1665.27
KWh/m2	117.33	2.32E+02	2.25E+01	2.42E+02	2.38E+02	4.23E+01

thermal zone 23	thermal zone 24	thermal zone 25	thermal zone 27	thermal zone 28	thermal zone 29
8073.876604	7270.073006	5884.923719	4842.830105	1774.203962	4138.412424
7.98E+02	1.18E+03	3.41E+03	1.75E+03	1.49E+03	1.54E+03
9963.946591	2616.582613	0	2093.26609	0	1569.949568
904.2909511	753.5757926	37.67878963	602.8606341	37.67878963	452.1454755
19740.20	11817.13	9330.01	9287.38	3301.26	7696.10
2.37E+02	1.31E+02	4.96E+01	1.24E+02	7.34E+01	1.28E+02

Το τελευταίο βήμα της ανάλυσης των δεδομένων από το Open Studio είναι η ταξινόμηση των θερμικών ζωνών με βάση την κατανάλωσή τους, σε φθίνουσα σειρά. Αυτή η ταξινόμηση είναι αυτή που αξιοποιείται από την εφαρμογή για την χρήση ενεργειακής πολιτικής (energy policy). Αν ο χρήστης επιλέξει την κατηγορία saving ή critical τότε αυτόματα ενεργοποιείται η αντίστοιχη πολιτική. Σε συνδυασμό με το πεδίο “Thermal zone level”, γίνεται η επιλογή των ζωνών με την

μεγαλύτερη κατανάλωση ενέργειας επιλέγοντας από το 1 έως το 12 βάσει των αναγκών που θέτει ο διαχειριστής του κτιρίου.

ΠΙΝΑΚΑΣ 3. Κατάταξη θερμικών ζωνών

Rank (descending order)		
thermal zone 20	242.0687589	kWh
thermal zone 21	238.0806194	kWh
thermal zone 23	237.3761887	kWh
thermal zone 18	231.8341154	kWh
thermal zone 24	131.1556814	kWh
thermal zone 29	127.7360607	kWh
thermal zone 27	123.6668787	kWh
thermal zone 17	117.3315187	kWh
thermal zone 28	73.36127319	kWh
thermal zone 25	49.5960624	kWh
thermal zone 22	42.33008985	kWh
thermal zone 19	22.54080013	kWh

#### 4.7 Ανάπτυξη κανόνων στον Gorgia

Το κύριο μέρος της εφαρμογής αποτελεί το σύνολο των αρχείων με τους κανόνες και τις πολιτικές που δημιουργήθηκαν στον Gorgia. Για την κάλυψη των διαφορετικών αναγκών που προέκυψαν από τις χρήσεις του κτιρίου K2, δημιουργήθηκαν 3 διαφορετικές πολιτικές. Η πρώτη πολιτική αφορά τον χρήστη και τις βασικές λειτουργίες που μπορεί να χειριστεί. Η δεύτερη πολιτική αφορά την διαχείριση ενέργειας και περιλαμβάνει ένα σύνολο κανόνων που στόχο έχουν την εφαρμογή μιας πολιτικής εξοικονόμησης ενέργειας βάσει των αναγκών που υπάρχουν κάθε φορά. Η τελευταία πολιτική αφορά τις καταστάσεις έκτακτης ανάγκης και τη διαχείριση των συσκευών μέσα σε αυτές. Οι πολιτικές αυτές έχουν η κάθε μια την δικιά της προτεραιότητα η οποία έχει δηλωθεί μέσω του αρχείου “Meta data policy”. Με αυτό τον τρόπο είναι ξεκάθαρη η λειτουργία της εκάστοτε πολιτικής όταν αυτή έρχεται σε σύγκρουση με μια άλλη. Επιπλέον έχει δημιουργηθεί ένα αρχείο (background) το οποίο περιέχει το σύνολο των δεδομένων που παραμένουν αμετάβλητα κατά την εκτέλεση της εφαρμογής. Τέλος όλα τα αρχεία με τις πολιτικές έχουν συνδυαστεί σε ένα συγκεντρωτικό αρχείο (combined data) που εμπεριέχει όλους του κανόνες για την λειτουργία της εφαρμογής. Παρακάτω αναλύονται τα χαρακτηριστικά του κάθε ενός από τα αρχεία πολιτικών που δημιουργήθηκαν:

#### 4.7.1 User's policy

Το πρώτο σύνολο κανόνων αφορούν την πολιτική του χρήστη. Πρόκειται για μια ευρύτερη πολιτική που χρησιμοποιείται σαν κυρίαρχη πολιτική όταν δεν υπερκαλύπτεται από κάποια άλλη με μεγαλύτερη προτεραιότητα. Ο χρήστης μπορεί να ενεργοποιήσει ή να απενεργοποιήσει την συσκευή που επιθυμεί. Επίσης μπορεί να επιλέξει ανάμεσα στην λειτουργία ψύξης ή θέρμανσης. Τέλος έχει την δυνατότητα να θέσει την επιθυμητή θερμοκρασία στον χώρο. Είναι σημαντικό να αναφέρουμε ότι η άρνηση ενός κατηγορήματος (negation) που αναλύονται παρακάτω, μπορεί να δημιουργήσει σύγκρουση μεταξύ των κανόνων. Για την επίλυση αυτών των συγκρούσεων θέτονται οι προτεραιότητες που θα δούμε στην συνέχεια.

```
rule(r11(Room, Device), switchOn(Room, Device), []):-  
isOff(Device,Room),tempOutOfBounds(Room),device(Device,heater_cooler,Room).  
rule(r12(Room, Device), neg(switchOn(Room, Device)), []):-  
isOff(Device,Room),device(Device,heater_cooler,Room).  
rule(r21(Room, Device), neg(switchOff(Room, Device)), []):-  
isOn(Device,Room),tempOutOfBounds(Room),device(Device,heater_cooler,Room).  
rule(r22(Room, Device), switchOff(Room, Device), []):-  
isOn(Device,Room),device(Device,heater_cooler,Room).
```

Επίσης η πολιτική αυτή περιλαμβάνει μια σειρά από κανόνες προτίμησης βάσει των οποίων γίνεται προτεραιοποίηση μεταξύ των διαφορετικών επιλογών. Οι προτιμήσεις μεταξύ των κανόνων περιγράφονται στον παρακάτω κώδικα.

```
rule(p11(Room, Device), prefer(r11(Room, Device),r12(Room, Device)), []).  
rule(p12(Room, Device), prefer(r12(Room, Device),r11(Room, Device)), []):-  
notWithinHours(Room).  
rule(c12(Room, Device), prefer(p12(Room, Device),p11(Room, Device)), []).  
rule(p21(Room, Device), prefer(r21(Room, Device),r22(Room, Device)), []).  
rule(p22(Room, Device), prefer(r22(Room, Device),r21(Room, Device)), []):-  
notWithinHours(Room).  
rule(c22(Room, Device), prefer(p22(Room, Device),p21(Room, Device)), []).
```

#### 4.7.2 Energy management policy

Πρόκειται για την πολιτική που ενεργοποιείται σε περίπτωση που ο διαχειριστής επιθυμεί να μειώσει την ενεργειακή κατανάλωση του κτιρίου. Βασικός κανόνας για την εφαρμογή αυτής της πολιτικής είναι ο αισθητήρας κίνησης ο οποίος απενεργοποιεί τις συσκευές στον χώρο εφόσον δεν υπάρχει κίνηση.

```
rule(r_mose(Room, Device), switchOff(Room, Device), []):-  
noMotion(Room),isOn(Device,Room),device(Device,heater_cooler,Room).
```

Επιπλέον ο διαχειριστής του κτιρίου μπορεί να επιλέξει μεταξύ δύο διαφορετικών επιπέδων αναφορικά με την εξοικονόμηση ενέργειας. Το πρώτο επίπεδο, “saving mode”, θέτει την



θερμοκρασία στις διαθέσιμες συσκευές εντός συγκεκριμένων ορίων (SavingBounds). Τα όρια αυτά είναι 19 C° για την λειτουργία της θέρμανσης και 26 C° για την λειτουργία της ψύξης. Αυτές οι δύο τιμές έχουν χρησιμοποιηθεί στην μοντελοποίηση του κτιρίου στο Open Studio βάσει των οποίων έγινε το simulation της ενεργειακής κατανάλωσης του κτιρίου για ένα ολόκληρο χρόνο.

```
rule(r11(Room, Device), switchOn(Room, Device), []):-
isOff(Device,Room),savingMode(Room),tempOutOfSavingBounds(Room),device(Device,heater_
cooler,Room).
rule(r12(Room, Device), neg(switchOn(Room, Device)), []):-
isOff(Device,Room),savingMode(Room),device(Device,heater_cooler,Room).
rule(p11(Room, Device), prefer(r11(Room, Device),r12(Room, Device)), []).
rule(p12(Room, Device), prefer(r12(Room, Device),r11(Room, Device)), []):-
notWithinHours(Room).
rule(c12(Room, Device), prefer(p12(Room, Device),p11(Room, Device)), []).
```

```
rule(r21(Room, Device), neg(switchOff(Room, Device)), []):-
isOn(Device,Room),savingMode(Room),tempOutOfSavingBounds(Room),device(Device,heater_
cooler,Room).
rule(r22(Room, Device), switchOff(Room, Device), []):-
isOn(Device,Room),savingMode(Room),device(Device,heater_cooler,Room).
rule(p21(Room, Device), prefer(r21(Room, Device),r22(Room, Device)), []).
rule(p22(Room, Device), prefer(r22(Room, Device),r21(Room, Device)), []):-
notWithinHours(Room).
rule(c22(Room, Device), prefer(p22(Room, Device),p21(Room, Device)), []).
```

Το δεύτερο επίπεδο “critical mode”, ενεργοποιείται στην περίπτωση που υπάρχει πολύ αυξημένη κατανάλωση ενέργειας και ο διαχειριστής επιθυμεί να λάβει δραστικά μέτρα. Σε αυτή την περίπτωση όλες οι συσκευές που βρίσκονται ανάμεσα στα δωμάτια με την υψηλή κατανάλωση απενεργοποιούνται.

```
rule(r_crmo(Room, Device), switchOff(Room, Device), []):-
criticalMode(Room),isOn(Device,Room),device(Device,heater_cooler,Room).
```

#### 4.7.3 Emergency policy

Η πολιτική αυτή αναφέρεται στις καταστάσεις έκτακτης ανάγκης. Σε περίπτωση που υπάρξει υψηλή τιμή διοξειδίου του άνθρακα τότε ενεργοποιείται ο συναγερμός στο χώρο ενώ ταυτόχρονα οι συσκευές για λόγους ασφαλείας απενεργοποιούνται και δεν είναι δυνατόν να τεθούν εκ νέου σε λειτουργία εφόσον δεν λήξει ο συναγερμός.

```
rule(r_em1(Device, Room), switchOff(Room, Device),[]):-isOn(Device,Room),alert(Room).
rule(r_em4(Device, Room), neg(switchOff(Room, Device)),[]):-
isOn(Device,Room),alert(Room),device(Device,alarm,Room).
rule(r_em2(Device, Room), neg(switchOn(Room, Device)),[]):-isOff(Device,Room),alert(Room).
rule(r_em3(Device, Room), switchOn(Room,Device),[]):-
isOff(Device,Room),alert(Room),device(Device,alarm,Room).
```

#### 4.7.4 Combined data

Αυτό το αρχείο προέκυψε χρησιμοποιώντας την μεθοδολογία όπως αυτή παρουσιάζεται στο paper «Towards Multipolicy Argumentation» [7]. Πρόκειται για ένα συνδυαστικό αρχείο που εμπεριέχει όλες τις επιμέρους πολιτικές που έχουν χρησιμοποιηθεί συνδυασμένες με την meta-policy, η οποία αναλύεται στην συνέχεια. Ταυτόχρονα στην αρχή αυτού, γίνεται κλήση του Background file.

#### 4.7.5 Meta data policy

Αυτό το αρχείο χρησιμοποιείται για καθορισμό των προτεραιοτήτων μεταξύ των διαφορετικών πολιτικών που έχουν χρησιμοποιηθεί. Οι τρεις πολιτικές που έχουν αναπτυχθεί είναι:

- Personal: Αφορά τις επιλογές του χρήστη αναφορικά με την θερμοκρασία που επιθυμεί στον χώρο εργασίας όπως διαμορφώνεται μέσω του αρχείου “User’s policy”.
- Emergency: Αφορά τις διαδικασίες που ενεργοποιούνται σε έκτακτες περιπτώσεις όπως αυτές διαμορφώθηκαν στο αρχείο “Emergency policy”.
- Management: Αφορά τις διαδικασίες που ενεργοποιούνται στην περίπτωση που ο διαχειριστής ενεργοποιήσει την λειτουργία εξοικονόμησης ενέργειας. Η πολιτική αυτή αναφέρεται στο αρχείο “Energy management policy”.

Προκειμένου λοιπόν να αποφευχθούν πιθανές συγκρούσεις μεταξύ αυτών των πολιτικών γίνεται προτεραιοποίηση με βάση την σημαντικότητα της κάθε μιας.

```
rule(r1(personal), policy(personal), []).
```

```
rule(r2(management), policy(management), []).
```

```
rule(r3(emergency), policy(emergency), []).
```

```
rule(p1(emergency), prefer(r3(emergency), r1(personal)), []).
```

```
rule(p2(emergency), prefer(r3(emergency), r2(management)), []).
```

```
rule(p3(management), prefer(r2(management), r1(personal)), []).
```

```
rule(p4(personal), prefer(r1(personal), r2(management)), []):-
```

```
device(Device,cooler,Room),computerRoom(Room).
```

```
rule(c1(personal), prefer(p4(personal), p3(management)), []).
```

Η πολιτική έκτακτης ανάγκης (emergency) διαθέτει προτεραιότητα έναντι των άλλων δύο πολιτικών. Επίσης η πολιτική της ενεργειακή διαχείρισης (management) διαθέτει μεγαλύτερη προτεραιότητα από την πολιτική του χρήστη (personal). Εξαιρέση σε αυτόν τον κανόνα υπάρχει μόνο στην περίπτωση που χώρος για τον οποίο εφαρμόζεται ο κανόνας είναι Computer Room. Σε αυτή την περίπτωση η πολιτική personal έχει προτεραιότητα έναντι της πολιτικής management. Αυτό συμβαίνει καθώς το computer room είναι ένα χώρος με ιδιαίτερες ανάγκες σε ψύξη καθ’ όλη

την διάρκεια της μέρας και δεν είναι δυνατόν να επηρεάζεται από άλλες πολιτικές, πέραν βέβαια της περίπτωσης έκτακτης ανάγκης.

#### 4.7.6 Background file

Το αρχείο αυτό καλείται μέσω της πολιτικής "combined data", στην οποία γίνεται ο συνδυασμός του συνόλου των πολιτικών με τις μετά-πολιτικές, σύμφωνα με την διαδικασία που περιγράφεται στο «Towards Multipolicy Argumentation» [7]. Το αρχείο αυτό εμπεριέχει το σύνολο των παραμέτρων που παραμένουν αμετάβλητες κατά την εκτέλεση της εφαρμογής μας.

Στην αρχή ορίζονται οι διαφορετικοί τύποι δωματίων όπως: auditorium, office (sensor room) και computer room. Ο κάθε ένας από αυτούς τους χώρους διαθέτει ξεχωριστή συσκευή ψύξης, θέρμανσης, αισθητήρα κίνησης καθώς και συναγερμό.

```
%space_121 auditorium
auditorium(space_121).
schedule(space_121,900,1800).
device(cooler1,cooler,space_121).
device(heater1,heater,space_121).
device(sensor1,sensor,space_121).
device(alarm1,alarm,space_121).
```

```
%space_110 user SENSOR ROOM
device(cooler2,cooler,space_110).
device(heater2,heater,space_110).
device(sensor2,sensor,space_110).
device(alarm2,alarm,space_110).
```

```
%space_115 computer room
computerRoom(space_115).
device(cooler3,cooler,space_115).
device(heater3,heater,space_115).
device(sensor3,sensor,space_115).
device(alarm3,alarm,space_115).
```

Στην συνέχεια γίνεται αντιστοίχιση του συνόλου των διαφορετικών χώρων που διαθέτουμε, στις ενεργειακές ζώνες που ανήκει ο κάθε ένας. Υπάρχουν συνολικά 29 χώροι (spaces) που αντιστοιχούν σε 12 διαφορετικές θερμικές ζώνες (thermal zones).

```
roomZone(space_121,thermal_Zone_23).
roomZone(space_110,thermal_Zone_20).
roomZone(space_115,thermal_Zone_18).
roomZone(space_112,thermal_Zone_20).
roomZone(space_111,thermal_Zone_20).
roomZone(space_114,thermal_Zone_18).
```

```

roomZone(space_113,thermal_Zone_18).
roomZone(space_120,thermal_Zone_23).
roomZone(space_122,thermal_Zone_21).
roomZone(space_123,thermal_Zone_21).
roomZone(space_125,thermal_Zone_22).
roomZone(space_133,thermal_Zone_17).
roomZone(space_134,thermal_Zone_25).
roomZone(space_137,thermal_Zone_24).
roomZone(space_147,thermal_Zone_27).
roomZone(space_145,thermal_Zone_28).
roomZone(space_148,thermal_Zone_27).
roomZone(space_162,thermal_Zone_24).
roomZone(space_151,thermal_Zone_30).
roomZone(space_159,thermal_Zone_17).
roomZone(space_160,thermal_Zone_17).
roomZone(space_161,thermal_Zone_24).
roomZone(space_163,thermal_Zone_24).
roomZone(space_164,thermal_Zone_24).
roomZone(space_165,thermal_Zone_27).
roomZone(space_166,thermal_Zone_27).
roomZone(space_169,thermal_Zone_29).
roomZone(space_171,thermal_Zone_29).
roomZone(space_170,thermal_Zone_29).
roomZone(space_109,thermal_Zone_19).

```

Όπως έχουμε ήδη περιγράψει προκειμένου να γίνει αποτελεσματική διαχείριση της λειτουργίας εξοικονόμησης ενέργειας, οι θερμικές ζώνες κατατάσσονται σε φθίνουσα σειρά ξεκινώντας από αυτή με την μεγαλύτερη κατανάλωση.

```

level(thermal_Zone_20,1).
level(thermal_Zone_21,2).
level(thermal_Zone_23,3).
level(thermal_Zone_18,4).
level(thermal_Zone_24,5).
level(thermal_Zone_29,6).
level(thermal_Zone_27,7).
level(thermal_Zone_17,8).
level(thermal_Zone_28,9).
level(thermal_Zone_25,10).
level(thermal_Zone_22,11).
level(thermal_Zone_19,12).

```

Αφού έχουμε ορίσει τις συσκευές, τους χώρους και τις ενεργειακές ζώνες πρέπει να δηλώσουμε μια σειρά από κανόνες που αφορούν τις διαφορετικές πολιτικές. Οι κανόνες αυτοί είναι:

- **General rules**

Εδώ ορίζουμε για μεγαλύτερη ευκολία το είδος των συσκευών που υπάρχουν στον χώρο στην περίπτωση που αυτές είναι συσκευές ψύξης (cooler) ή θέρμανσης (heating).

```
device(Y,heater_cooler,X):-device(Y,cooler,X).  
device(Y,heater_cooler,X):-device(Y,heater,X).
```

- **Energy management rules**

Πρόκειται για τους κανόνες που αφορούν την ενεργειακή πολιτική του κτιρίου. Συγκεκριμένα γίνεται χρήση του αισθητήρα κίνησης ο οποίος παίρνει την τιμή false σε περίπτωση που δεν υπάρχει παρουσία στον χώρο. Επιπλέον βλέπουμε την λειτουργία saving mode. Σε περίπτωση που ενεργοποιηθεί, τότε όλοι οι χώροι που βρίσκονται μέσα στις επιλεγμένες θερμικές ζώνες, επανέρχονται στις προεπιλεγμένες τιμές ψύξης και θέρμανσης. Τέλος η λειτουργία critical mode σβήνει όλες τις συσκευές που βρίσκονται μέσα στις θερμικές ζώνες που έχει επιλέξει ο χρήστης.

```
noMotion(Room):- device(Dev,Type,Room),Type=sensor,reading(Dev,false).  
savingMode(Room) :-  
device(Dev,heater_cooler,Room),roomZone(Room,Zone),level(Zone,L),limit(AL,saving),L  
<AL.  
tempOutOfSavingBounds(Room) :- hasTemp(Room,  
Temp),limitByUser(Room,_,Type),Temp>26,Type=cool.  
tempOutOfSavingBounds(Room) :- hasTemp(Room,  
Temp),limitByUser(Room,_,Type),Temp<19,Type=heat.  
criticalMode(Room):-  
device(Dev,heater_cooler,Room),roomZone(Room,Zone),level(Zone,L),limit(AL,critical),L  
<AL.
```

- **Emergency rules**

Οι κανόνες που αφορούν τις καταστάσεις έκτακτης ανάγκης αναφέρονται στην περίπτωση ύπαρξης μέτρησης διοξειδίου του άνθρακα που υπερβαίνει το όριο. Το όριο αυτής της μέτρησης είναι 2000 Vpm (Volume per million) όπως ορίζεται από την νομοθεσία. Σε περίπτωση που ξεπεραστεί αυτό το όριο ενεργοποιείται ο συναγερμός στον συγκεκριμένο χώρο.

```
highCO2(Room):-co2Level(Room,X),X>2000.  
alert(Room):-highCO2(Room).
```

- **User rules**

Αναφέρονται στους κανόνες του χρήστη, ο οποίος έχει την δυνατότητα να επιλέξει την επιθυμητή θερμοκρασία για τον χώρο στον οποίο βρίσκεται επιλέγοντας ταυτόχρονα την λειτουργία ψύξης ή θέρμανσης. Επιπλέον σε αυτό το σημείο ορίζεται και ο κανόνας “notWithinHours”, ο οποίος ενεργοποιείται όταν η ώρα η οποία έχει επιλεγεί είναι εκτός του χρονοδιαγράμματος βάσει του οποίου λειτουργεί το αμφιθέατρο.

```
notWithinHours(Room):-  
current_time(X),auditorium(Room),schedule(Room,START,FINISH),X<START.  
notWithinHours(Room):-  
current_time(X),auditorium(Room),schedule(Room,START,FINISH),X>FINISH.
```

```
tempOutOfBounds(Room) :- hasTemp(Room, Temp),  
limitByUser(Room,X,Type),Temp>X,Type=cool.  
tempOutOfBounds(Room) :- hasTemp(Room, Temp),  
limitByUser(Room,X,Type),Temp<X,Type=heat.
```

## Κεφάλαιο 5<sup>ο</sup>

### Πιλοτική εφαρμογή

#### 5.1 Το Πολυτεχνείο Κρήτης

Το Πολυτεχνείο Κρήτης [17] αποτελεί ένα Ανώτατο Εκπαιδευτικό Ίδρυμα της Ελλάδας. Βρίσκεται στην θέση Ακρωτήρι στον Νομό Χανίων και εκτείνεται σε μια έκταση 2900 στρεμμάτων. Διαθέτει 5 σχολές:

- Σχολή Μηχανικών Παραγωγής και Διοίκησης
- Σχολή Μηχανικών Ορυκτών Πόρων
- Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
- Σχολή Χημικών Μηχανικών και Μηχανικών Περιβάλλοντος
- Σχολή Αρχιτεκτόνων Μηχανικών



ΕΙΚΟΝΑ 11. Αεροφωτογραφία της πολυτεχνειούπολης

Συγκεκριμένα για τις ανάγκες την παρούσας διπλωματικής ασχολούμαστε με την σχολή Χημικών Μηχανικών και Μηχανικών Περιβάλλοντος η οποία καταλαμβάνει τρία κτίρια (K1,K2,K3) συνολικής επιφάνειας 3.000 m<sup>2</sup>. Το κτίριο το οποίο αναλύεται στην παρούσα διπλωματική είναι το K2.

#### 5.2 Κτίριο K2 της Σχολής Μηχανικών Περιβάλλοντος

Το K2 αποτελεί το κύριο κτίριο της σχολής ΜΗΠΕΡ καθώς στον 1<sup>ο</sup> όροφο αυτού στεγάζεται η γραμματεία της σχολής αλλά και πλήθος γραφείων καθηγητών. Στο ισόγειο του κτιρίου βρίσκεται το Μηχανογραφικό Κέντρο του Πολυτεχνείου Κρήτης. Σε αυτό βρίσκονται πέντε αίθουσες



διδασκαλίας με συνολικά 154 σταθμούς εργασίας με ηλεκτρονικούς υπολογιστές για την πραγματοποίηση εργαστηριακών μαθημάτων. Επιπλέον βρίσκονται τοποθετημένοι 6 επαγγελματικοί εκτυπωτές που καλύπτουν τις ανάγκες των μελών της πολυτεχνειακής κοινότητας

Η συνολική χωρητικότητα του κτιρίου είναι 1570 τετραγωνικά μέτρα τα οποία κατανέμονται σε τρία επίπεδα. Το πρώτο επίπεδο, ισόγειο, περιλαμβάνει οχτώ εργαστήρια και τέσσερις κοινόχρηστους χώρους. Το δεύτερο επίπεδο, πρώτος όροφος, περιλαμβάνει δεκαπέντε γραφεία καθηγητών και δύο κοινόχρηστους χώρους. Στο τελευταίο επίπεδο, δεύτερος όροφος, αποτελείται από έναν μεγάλο ενιαίο χώρο εμβαδού 533 τετραγωνικών μέτρων που είναι η σοφίτα.



ΕΙΚΟΝΑ 12. Το κτίριο K2 της Σχολής Μηχανικών Περιβάλλοντος

Συνολικά οι τριάντα χώροι που υπάρχουν στο κτίριο για τις ανάγκες της μοντελοποίησης χωρίστηκαν σε τέσσερις κατηγορίες ανάλογα με το είδος:

- Εργαστήρια
- Γραφεία
- Κοινόχρηστοι χώροι
- Σοφίτα



### 5.3 Χρήση με πραγματικά δεδομένα

Στο κτίριο των Χημικών Μηχανικών και Μηχανικών Περιβάλλοντος του Πολυτεχνείου Κρήτης και συγκεκριμένα στο “Εργαστήριο Δομημένου Περιβάλλοντος και Διαχείρισης Ενέργειας” , που βρίσκεται στον 1<sup>ο</sup> όροφο του κτιρίου, υπάρχουν τοποθετημένοι μια σειρά από αισθητήρες τύπου KNX. Οι συγκεκριμένοι αισθητήρες καλύπτουν ένα εύρος δεδομένων και μετρήσεων που αξιοποιούνται για την δοκιμή του προγράμματος και την λήψη αποφάσεων. Σημαντικό στοιχείο στην επιλογή των αισθητήρων ήταν να εξεταστούν οι παράμετροι που έχουν το μεγαλύτερο βάρος στην δημιουργία πολιτικών και να οδηγούν στην σωστότερη λήψη αποφάσεων. Συνεπώς για αυτό τον λόγο επιλεχθήκαν αισθητήρες που μετρούν την ποιότητα του περιβάλλοντος, την κατανάλωση ρεύματος, την ποσότητα φωτός στον χώρο καθώς και κατάλληλοι server που εντάσσονται στο δίκτυο των αισθητήρων και μας βοηθούν με την εξαγωγή των δεδομένων. Για αυτόν ακριβώς τον λόγο επιλέχθηκαν οι παρακάτω αισθητήρες που λειτουργούν στο πρότυπο KNX και οι οποίοι παραμετροποιήθηκαν με την χρήση του προγράμματος ETS5:

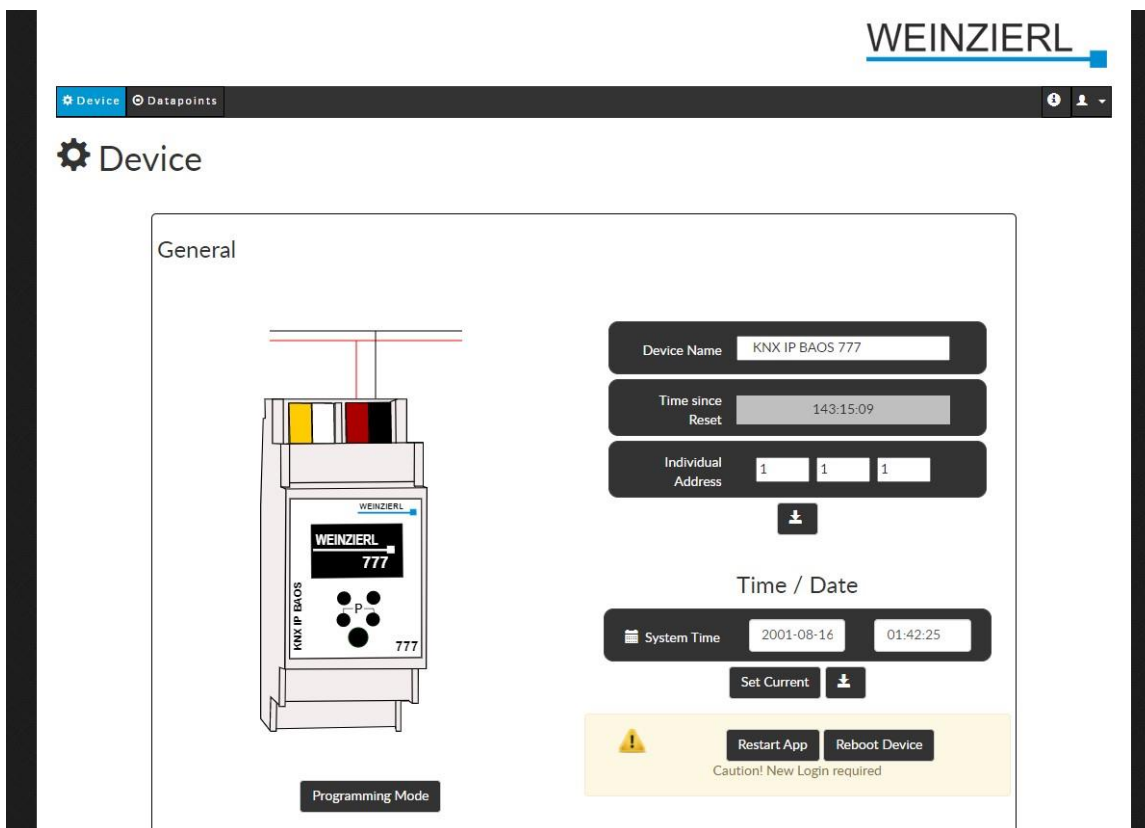
- **KNX IP BAOS 777** [18]: Αποτελεί τον server πάνω στον οποίο συνδέονται οι υπόλοιποι αισθητήρες. Τα αρχικά BAOS σημαίνουν “Bus Access and Object Server”. Επιτρέπει την σύνδεση μέχρι και 10 αισθητήρων ταυτόχρονα στο δίκτυό του. Η σύνδεση στο διαδίκτυο γίνεται μέσω ethernet με ταχύτητες μέχρι 100 Mbit/s. Η πρόσβαση σε αυτόν γίνεται μέσω μιας IP address που διαθέτει και του ψηφιακού περιβάλλοντος που έχει δημιουργηθεί από τον κατασκευαστή. Η τροφοδοσία γίνεται με εξωτερική σύνδεση 12-30 V συνεχούς ρεύματος ή μέσω Ethernet καλωδίου (Power over Ethernet PoE), που αποτελεί και τον τρόπο τροφοδοσίας του στην παρούσα εργασία. Τέλος διαθέτει οθόνη OLED με 4 κουμπιά για την ρύθμιση των βασικών λειτουργιών του.
- **ABB i-bus KNX Energy Module** [19]: Πρόκειται για ένα ψηφιακό μετρητή τάσης και κατανάλωσης ενέργειας. Επιτρέπει την σύνδεση σε αυτόν έως και 3 συσκευών για τις οποίες μπορεί να μετράει την κατανάλωση και λειτουργεί σαν διακόπτης για αυτές τις συσκευές. Η ακρίβεια που δίνει για την κατανάλωση ενέργειας είναι της τάξης του  $\pm 6\%$  και για την τάση  $\pm 1\%$ . Μπορεί να μετρήσει κατανάλωση έως 4.600 W , σε εναλλασσόμενο ρεύμα έντασης έως 20A και τάσης έως 265V. Η τροφοδοσία του συγκεκριμένου αισθητήρα γίνεται με συνεχές ρεύμα τάσης 21-30V.
- **ABB i-bus® KNX Air Quality Sensor** [20]: Πρόκειται για έναν αισθητήρα με πολλαπλές δυνατότητες καθώς μπορεί να μετρήσει 3 διαφορετικές τιμές. Αυτές είναι η υγρασία, η θερμοκρασία και η ποσότητα διοξειδίου του άνθρακα (CO<sub>2</sub>) στον αέρα. Η ακρίβεια που διαθέτει είναι  $\pm 300$  ppm για την μέτρηση του CO<sub>2</sub>,  $\pm 5\%$  για την υγρασία και  $\pm 2\%$  για την θερμοκρασία. Επιπλέον διαθέτει δύο λυχνίες LED, μια για την παρουσία διοξειδίου του άνθρακα στον χώρο και μια για την σχετική υγρασία. Οι δύο λυχνίες αυτές μπορούν να

αλλάζουν μεταξύ 4 χρωμάτων (μπλε, πράσινο, κίτρινο, κόκκινο) ανάλογα με την παραμετροποίηση που θα κάνει ο χρήστης.

- **ABB i-bus® KNX Movement and Presence Detector Mini Basic** [21]: Ο συγκεκριμένος αισθητήρας μπορεί να ελέγξει την παρουσία στον χώρο. Τοποθετείται στον τοίχο σε ύψος 2,5 έως 4 μέτρα και μπορεί να ανιχνεύσει παρουσία σε ένα μετωπικό εύρος μετάδοσης 8 μέτρων. Επιπλέον μπορεί να ελέγξει την στάθμη φωτεινότητας και να ενεργοποιηθεί (On) ή να απενεργοποιηθεί (Off) αυτόματα ανάλογα με το κατώφλι που έχει ορίσει ο χρήστης. Το όριο φωτεινότητας που διαθέτει είναι μεταξύ 1-1000 lux.

#### 5.4 Δεδομένα από το KNX BAOS

Ο διάυλος επικοινωνίας KNX BAOS 777, αποτελεί τον server πάνω στον οποίο είναι συνδεδεμένοι όλοι οι υπόλοιποι αισθητήρες που διαθέτουμε και το μέσο λήψης των δεδομένων που λαμβάνουμε από αυτούς. Συγκεκριμένα το σύστημα διαθέτει μια μοναδική IP address στην οποία έχουμε πρόσβαση μέσω των στοιχείων σύνδεσης που μας παρέχει ο κατασκευαστής του. Εισερχόμενοι στην κεντρική σελίδα του συστήματος μπορούμε να κάνουμε κάποιες παραμετροποιήσεις, όπως την αλλαγή του ονόματος της συσκευής, τον ορισμό της ώρας αλλά και να επαναφέρουμε τις εργοστασιακές ρυθμίσεις σε περίπτωση που έχουμε κάποιο σφάλμα.



ΕΙΚΟΝΑ 13. Κεντρικό μενού επιλογών του BAOS 777

Επιπλέον διαθέτουμε την επιλογή να αλλάξουμε την IP address στην οποία βρισκόμαστε, τον κόμβο (gateway) αλλά και το υποδίκτυο. Στην συγκεκριμένη περίπτωση αυτές οι ρυθμίσεις έγιναν με βάση το δίκτυο του Πολυτεχνείου.

## Network

MAC Address	00:24:6D:01:2C:9B	IP Assignment	Manual ▼
IP Address	147.27.33.130	Subnet Mask	255.255.255.0
Default Gateway	147.27.33.254		

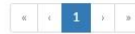
⬇

ΕΙΚΟΝΑ 14. Ρυθμίσεις Δικτύου BAOS 777

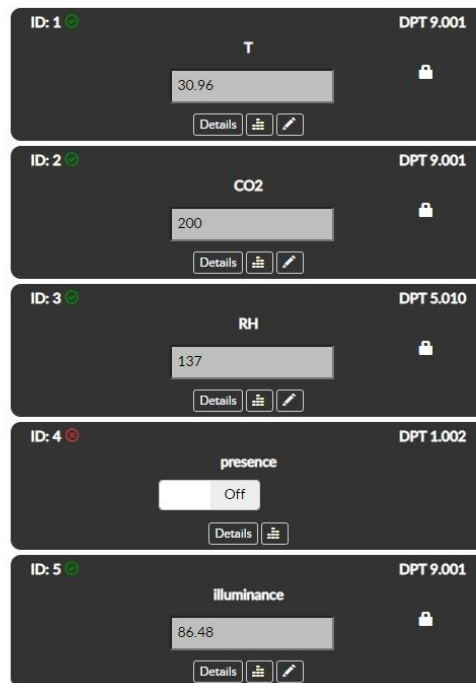
Στην δεύτερη καρτέλα που υπάρχει στο σύστημα, βρίσκονται οι συσκευές οι οποίες είναι συνδεδεμένες πάνω στο BAOS 777. Όπως μπορούμε να δούμε και στην παρακάτω εικόνα,

υπάρχουν 4 διαφορετικές μετρήσεις τις οποίες παίρνουμε από τους αισθητήρες μας. Η πρώτη μέτρηση αφορά την θερμοκρασία σε βαθμούς κελσίου, η δεύτερη αφορά το διοξείδιο του άνθρακα στην ατμόσφαιρα το οποίο μετριέται σε VPM, η τρίτη μέτρηση αφορά την υγρασία στον χώρο σε κλίμακα επί της εκατό (%). Να σημειώσουμε σε αυτό το σημείο ότι η κλίμακα λόγω ρυθμίσεων βρίσκεται σε εύρος 0-255, δηλαδή το 100% αντιστοιχεί στην τιμή 255. Η τελευταία μέτρηση είναι η φωτεινότητα που υπάρχει στον χώρο. Η μέτρηση αυτής γίνεται σε lux με μέγιστη τιμή το 1000.

## ⦿ Datapoints



Page: 1 / 1 - 25 Datapoints per page



ΕΙΚΟΝΑ 15. Συνδεδεμένες συσκευές στο BAOS 777

## 5.5 Εφαρμογή σεναρίων

Προκειμένου να ελέγχουμε την αποτελεσματικότητα του εργαλείου που αναπτύχθηκε, είναι σκόπιμο να εξεταστεί ένα σύνολο διαφορετικών σεναρίων που καλύπτουν όλο το πλήθος των πιθανών αποτελεσμάτων. Έχοντας ως παράδειγμα το κτίριο των Μηχανικών Περιβάλλοντος και με την βοήθεια των αισθητήρων θα προσομοιώσουμε διαφορετικές καταστάσεις που μπορεί να δημιουργηθούν. Τα αποτελέσματα αυτών θα δοθούν μέσω του Gorgia, ο οποίος συνδυάζει τα δεδομένα και λαμβάνει την απόφαση.

Επιπλέον στην επιλογή δωματίου βρίσκεται το δωμάτιο τύπου “γραφείο” (office). Σε αυτό το δωμάτιο βρίσκονται τοποθετημένοι και οι αισθητήρες που διαθέτουμε. Συνεπώς με την λήψη των δεδομένων από το KNX BAOS Rest API, συμπληρώνονται αυτόματα τα εξής κελιά: θερμοκρασία (Temperature), παρουσία στον χώρο (Motion), διοξείδιο του άνθρακα (Current CO2) καθώς και η ώρα (Current time). Αξίζει να σημειώσουμε ότι παρόλο που τα κελιά συμπληρώνονται αυτόματα με τα δεδομένα των αισθητήρων, ο χρήστης έχει την δυνατότητα να τα τροποποιήσει ανάλογα με την βούληση του.

Για την περιγραφή και την κατανόηση όλων των διαφορετικών πολιτικών και κανόνων εκτελέστηκαν συνολικά 10 σενάρια. Τα σενάρια αυτά εκτελέστηκαν μέσω του “Automated decision making tool” που έχει αναπτυχθεί και περιγράφονται αναλυτικά παρακάτω:

### Σενάριο 1°

Μέσα από το πάνελ επιλογών, θα επιλέξουμε το δωμάτιο τύπου “γραφείο” (office), ρυθμίζουμε την λειτουργία σε ψύξη (cool). Όλες οι συσκευές είναι απενεργοποιημένες (cooler, heater, alarm). Ρυθμίζουμε την επιθυμητή θερμοκρασία (Target Temperature) σε σημείο χαμηλότερο από την υπάρχουσα (Current Temperature). Ο αισθητήρας κίνηση έχει καταγράψει παρουσία στον χώρο. Το διοξείδιο του άνθρακα είναι στα 200 Vpm. Η ώρα είναι η υπάρχουσα, ενώ στο energy policy είναι απενεργοποιημένο στην λειτουργία none, όπως και το thermal zone level.

**Rooms**

**Action**

**Cooler** ☐

**Heater** ☐

**Alarm** ☐

**Current Temperature**

**Target Temperature**

**Motion** ☒

**Current CO2**

**Current Time**

**Energy policy**

**Thermal zone level**

## Results

Switch On cooler: true:According to the user policy, if the device (cooler, heater) is Off and the temperature is out of the bounds, then switch On the device - ['{E=[user\_r11(space\_110, cooler2, user), r1(space\_110, cooler2, user)]}']  
 Do not switch On cooler: false  
 Switch Off cooler: false  
 Do not switch Off cooler: false  
 Switch On alarm: false  
 Do not switch On alarm: false  
 Switch Off alarm: false  
 Do not switch Off alarm: false

EIKONA 16. Σενάριο 1

Τρέχοντας το πρόγραμμα λαμβάνουμε το αποτέλεσμα το οποίο είναι η εντολή ενεργοποίησης του συστήματος ψύξης (cooler). Οι κανόνες οι οποίοι ενεργοποιήθηκαν είναι οι:

*user\_r11(Room, Device, user), switchOn\_local(Room, Device, user), []):-isOff(Device, Room), tempOutOfBounds(Room), device(Device, heater\_cooler, Room)  
 rule(r1(Room, Device, Policy1), switchOn(Room, Device), [switchOn\_local(Room, Device, Policy1)]).*

Αυτοί οι κανόνες αφορούν την πολιτική του χρήστη (user) και οι προϋποθέσεις ενεργοποίησης τους ήταν: η θερμοκρασία να είναι εκτός των ορίων που έχει θέσει ο χρήστης, το σύστημα ψύξης είναι απενεργοποιημένο ενώ ταυτόχρονα υπάρχει παρουσία ανθρώπων στο χώρο.

Να σημειωθεί ότι οι επιμέρους πολιτικές που εμπεριέχονται στο αρχείο User's policy, συνδυάζονται μέσω των local πολιτικών που έχουν δημιουργηθεί αυτόματα (βλ. rule(user\_r11)). Ο ορισμός των σωστών κατηγορημάτων γίνεται μέσω των meta-policies. Τέλος ο κανόνας rule(user\_r11) έχει προκύψει από κανόνα που περιγράφεται στην βασική πολιτική του User:

rule(r11(Room, Device), switchOn(Room, Device), []):-  
 isOff(Device,Room),tempOutOfBounds(Room),device(Device,heater\_cooler,Room).

## Σενάριο 2°

Έχει επιλεγεί το δωμάτιο τύπου “γραφείο” (office), το σύστημα ψύξης είναι ενεργοποιημένο ενώ η θέρμανση και ο συναγερμός απενεργοποιημένα. Η θερμοκρασία στόχος βρίσκεται σε χαμηλότερα επίπεδα από την υπάρχουσα. Ο αισθητήρας κίνησης, έχει καταγράψει παρουσία στον χώρο. Το διοξείδιο του άνθρακα είναι στα 200 Vpm. Η ώρα είναι η υπάρχουσα, ενώ το energy policy είναι απενεργοποιημένο στην λειτουργία none, όπως και το thermal zone level.

Rooms	Office (Sensor) ▼
Action	Cool ▼
Cooler	<input checked="" type="checkbox"/>
Heater	<input type="checkbox"/>
Alarm	<input type="checkbox"/>
Current Temperature	32.3
Target Temperature	21
Motion	<input checked="" type="checkbox"/>
Current CO2	200
Current Time	11:11 AM 🕒
Energy policy	None ▼
Thermal zone level	1 ▼
<div>Prove</div>	

## Results

Switch On cooler: false  
Do not switch On cooler: false  
Switch Off cooler: false  
Do not switch Off cooler: true:According to the user policy,if the device (cooler,heater) is On, the temperature is out of bounds,then don't switch On the device - ['E=[user\_r21(space\_110, cooler2, user), r3(space\_110, cooler2, user)]']  
Switch On alarm: false  
Do not switch On alarm: false  
Switch Off alarm: false  
Do not switch Off alarm: false

### EIKONA 17. Σενάριο 2

Όπως μπορούμε να δούμε, οι απαντήσεις που λάβαμε είναι αρνητικές (false) πέραν από τον κανόνα που αποτρέπει το σβήσιμο του cooler, συνεπώς δεν θα πρέπει να πραγματοποιηθεί κάποια ενέργεια. Ο κανόνας που ενεργοποιήθηκε είναι ο:

```
user_r21(Room, Device, user), neg(switchOff_local(Room, Device, user)), []):-  
isOn(Device, Room), tempOutOfBounds(Room), device(Device, heater_cooler, Room)
```

Σύμφωνα με αυτόν βλέπουμε το επιχείρημα το οποίο έχει ενεργοποιηθεί προκειμένου να μην σβήσει ο cooler. Αυτός ο κανόνας ο οποίος αφορά την πολιτική του χρήστη (user) και οι προϋποθέσεις του ήταν: η θερμοκρασία να είναι εκτός των ορίων που έχει θέσει ο χρήστης, το σύστημα ψύξης είναι ενεργοποιημένο ενώ ταυτόχρονα υπάρχει παρουσία ανθρώπων στο χώρο.

Αντίστοιχα αποτελέσματα θα είχαμε, αν είχε επιλεγεί η λειτουργία θέρμανσης (heater).

### Σενάριο 3°

Το δωμάτιο που επιλέγουμε είναι τύπου “γραφείο” (office), ρυθμίζουμε την λειτουργία σε ψύξη (cool). Οι συσκευές heater, alarm είναι απενεργοποιημένες, ενώ ο cooler είναι ενεργοποιημένος. Ρυθμίζουμε την επιθυμητή θερμοκρασία (Target Temperature) σε σημείο χαμηλότερο από την υπάρχουσα (Current Temperature). Ο αισθητήρας κίνησης έχει καταγράψει παρουσία στον χώρο. Η ώρα είναι η υπάρχουσα, ενώ το energy policy είναι απενεργοποιημένο στην λειτουργία none, όπως και το thermal zone level. Σε αυτή την περίπτωση όμως η τιμή του διοξειδίου του άνθρακα είναι 2100 Vmp, που είναι μεγαλύτερη από το όριο των 2000 Vmp.

**Rooms**

**Action**

**Cooler** ☒

**Heater** ☐

**Alarm** ☐

**Current Temperature**

**Target Temperature**

**Motion** ☒

**Current CO2**

**Current Time**

**Energy policy**

**Thermal zone level**

## Results

Switch On cooler: false  
Do not switch On cooler: false  
Switch Off cooler: true:According to the emergency policy,if the device (cooler,heater) is On and there is an alert on the room, then switch Off the device - ['{E=[emergency\_r\_em1(space\_110, cooler2, emergency), r4(space\_110, cooler2, emergency)]}']  
Do not switch Off cooler: false  
Switch On alarm: true:According to the emergency policy,if the alarm is Off and there is an alert on the room, then switch On the alarm - ['{E=[emergency\_r\_em3(space\_110, alarm2, emergency), r1(space\_110, alarm2, emergency)]}']  
Do not switch On alarm: false  
Switch Off alarm: false  
Do not switch Off alarm: false

### EIKONA 18. Σενάριο 3

Τα αποτελέσματα που παρουσιάστηκαν πατώντας το prove δείχνουν ότι ο cooler πρέπει να απενεργοποιηθεί και ταυτόχρονα να ενεργοποιηθεί ο συναγερμός (alarm). Οι δύο κανόνες που ενεργοποιήθηκαν ήταν οι:

*emergency\_r\_em1(Room, Device, emergency), switchOff\_local(Room, Device, emergency), []):-isOn(Device, Room), alert(Room)*



*emergency\_r\_em3(Room, Device, emergency), switchOn\_local(Room, Device, emergency), []):-isOff(Device, Room), alert(Room),device(Device,alarm,Room)*

Οι δύο κανόνες αυτοί είναι της πολιτικής τύπου *emergency*. Ο πρώτος απενεργοποιεί τον cooler που είναι αναμμένος, εφόσον έχει σημάνει ο συναγερμός ανεξάρτητα από την παρουσία ανθρώπων στον χώρο και την θερμοκρασία που έχει θέσει ο χρήστης καθώς οι κανόνες *emergency* έχουν μεγαλύτερη προτεραιότητα σε σχέση με όλους τους άλλους τύπους (*user*, *energy management*). Ο δεύτερος κανόνας ενεργοποιεί τον συναγερμό του συστήματος καθώς υπάρχει *alert*, το οποίο έχει προκύψει λόγω της υψηλής τιμής του διοξειδίου του άνθρακα.

#### Σενάριο 4°

Το δωμάτιο που επιλέγουμε είναι τύπου “γραφείο” (*office*), ρυθμίζουμε την λειτουργία σε ψύξη (*cool*). Οι συσκευές *heater*, *alarm* είναι απενεργοποιημένες, ενώ ο *cooler* είναι ενεργοποιημένος. Ρυθμίζουμε την επιθυμητή θερμοκρασία (*Target Temperature*) σε υψηλότερο σημείο από την υπάρχουσα (*Current Temperature*). Ο αισθητήρας κίνησης έχει καταγράψει παρουσία στον χώρο. Η ώρα είναι η υπάρχουσα, ενώ το *energy policy* είναι απενεργοποιημένο στην λειτουργία *none*, όπως και το *thermal zone level*.

**Rooms**    
**Action**    
**Cooler** ☒   
**Heater** ☐   
**Alarm** ☐   
**Current Temperature**    
**Target Temperature**    
**Motion** ☒   
**Current CO2**    
**Current Time**    
**Energy policy**    
**Thermal zone level**

## Results

Switch On cooler: false  
 Do not switch On cooler: false  
 Switch Off cooler: true:According to the user policy,if the device (cooler,heater) is On, then switch Off the device - ['E=[user\_r22(space\_110, cooler2, user), r4(space\_110, cooler2, user)]']  
 Do not switch Off cooler: false  
 Switch On alarm: false  
 Do not switch On alarm: false  
 Switch Off alarm: false  
 Do not switch Off alarm: false

### EIKONA 19. Σενάριο 4

Εκτελώντας το πρόγραμμα παρατηρούμε ότι δίνεται η εντολή για το κλείσιμο του cooler, καθώς ενεργοποιήθηκε ο κανόνας:

*user\_r22(Room, Device, user), switchOff\_local(Room, Device, user), []):-isOn(Device, Room), device(Device, heater\_cooler, Room)*

Ο κανόνας αυτός ενεργοποιήθηκε καθώς ο cooler είναι ήδη ενεργοποιημένος αλλά η θερμοκρασία στον χώρο είναι μικρότερη από αυτή που έχει θέσει ο χρήστης. Συνεπώς ο cooler πρέπει να απενεργοποιηθεί.

### Σενάριο 5°

Το δωμάτιο που επιλέγουμε είναι τύπου “γραφείο” (office), ρυθμίζουμε την λειτουργία σε θέρμανση (heat). Οι συσκευές cooler, alarm είναι απενεργοποιημένες, ενώ ο heater είναι ενεργοποιημένος. Ρυθμίζουμε την επιθυμητή θερμοκρασία (Target Temperature) σε υψηλότερο σημείο από την υπάρχουσα (Current Temperature). Ο αισθητήρας κίνησης έχει καταγράψει

παρουσία στον χώρο. Η ώρα είναι η υπάρχουσα, ενώ το energy policy είναι απενεργοποιημένο στην λειτουργία none, όπως και το thermal zone level.

**Rooms**

**Action**

**Cooler** ☐

**Heater** ☒

**Alarm** ☐

**Current Temperature**

**Target Temperature**

**Motion** ☒

**Current CO2**

**Current Time**

**Energy policy**

**Thermal zone level**

## Results

Switch On heater: false  
Do not switch On heater: false  
Switch Off heater: false  
Do not switch Off heater: true:According to the user policy,if the device (cooler,heater) is On, the temperature is out of bounds,then don't switch On the device - ['!E=[user\_r21(space\_110, heater2, user), r3(space\_110, heater2, user)]]'  
Switch On alarm: false  
Do not switch On alarm: false  
Switch Off alarm: false  
Do not switch Off alarm: false

### EIKONA 20. Σενάριο 5

Εκτελώντας το πρόγραμμα παρατηρούμε ότι η μόνη εντολή δίνεται αφορά την αποτροπή από το κλείσιμο του heater καθώς ενεργοποιήθηκε ο κανόνας:

```
energy_r21(Room, Device, energy), neg(switchOff_local(Room, Device, energy)), []):-  
isOn(Device, Room), savingMode(Room), tempOutOfSavingBounds(Room),  
device(Device, heater_cooler, Room)
```

Ο κανόνας αυτός αφορά την πολιτική του user στην περίπτωση που η υπάρχουσα θερμοκρασία είναι εκτός των ορίων που έχει ορίσει ο χρήστης . Σε αυτή την περίπτωση, καθώς πρόκειται για θέρμανση, μας ενδιαφέρει να ισχύει: Current Temperature< Target Temperature. Συνεπώς σε συνδυασμό με την ύπαρξη ανθρώπων στον χώρο, ο heater δεν πρέπει να απενεργοποιηθεί.

### Σενάριο 6°

Το δωμάτιο που επιλέγουμε είναι τύπου “γραφείο” (office), ρυθμίζουμε την λειτουργία σε ψύξη (cool). Οι συσκευές heater, alarm είναι απενεργοποιημένες, ενώ ο cooler είναι ενεργοποιημένος. Ρυθμίζουμε την επιθυμητή θερμοκρασία (Target Temperature) σε χαμηλότερο σημείο από την

υπάρχουσα (Current Temperature). Ο αισθητήρας κίνησης έχει καταγράψει παρουσία στον χώρο. Η ώρα είναι η υπάρχουσα, και στο πεδίο του energy policy βάζουμε την τιμή Saving και το thermal zone level στο επίπεδο 12. Με αυτό τον τρόπο ενεργοποιούμε την λειτουργία εξοικονόμησης ενέργειας (saving mode) η οποία βάζει τις ενεργειακά κοστοβόρες ζώνες στις προεπιλεγμένες ρυθμίσεις. Οι προεπιλεγμένες ρυθμίσεις αφορούν συγκεκριμένες τιμές θερμοκρασίας ανάλογα με την λειτουργία που είναι ενεργοποιημένη. Συγκεκριμένα θέτουν την τιμή 26 C° όταν βρισκόμαστε στην λειτουργία cool και 19 C° όταν βρισκόμαστε στην λειτουργία heat. Με αυτό τον τρόπο επιτυγχάνεται μείωση στην κατανάλωση ενέργειας. Τέλος το επίπεδο thermal zone που έχει την τιμή 12, αναφέρεται στις ζώνες οι οποίες θα μουν στην λειτουργία saving. Το γραφείο αυτό ανήκει στην θερμικής ζώνη 20, η οποία βρίσκεται στην πρώτη θέση της κατάταξης των ζωνών με βάση την κατανάλωση τους. Εφόσον δηλαδή η τιμή που ορίζουμε στο Thermal zone level είναι μεγαλύτερη από την θέση κατάταξης που είναι το γραφείο, τότε μπαίνει στο saving mode. (Thermal zone level> Thermal zone ranking).

**Rooms**

**Action**

**Cooler** ☒

**Heater** ☐

**Alarm** ☐

**Current Temperature**

**Target Temperature**

**Motion** ☒

**Current CO2**

**Current Time**

**Energy policy**

**Thermal zone level**

## Results

```
Switch On cooler: false
Do not switch On cooler: false
Switch Off cooler: true:According to the energy managment policy;if the device (cooler,heater) is On and saving mode in On, then switch Off the
device - ['{E=[energy_r22(space_110, cooler2, energy), r4(space_110, cooler2, energy)]}']
Do not switch Off cooler: false
Switch On alarm: false
Do not switch On alarm: false
Switch Off alarm: false
Do not switch Off alarm: false
```

EIKONA 21. Σενάριο 6

Εκτελώντας το πρόγραμμα παρατηρούμε ότι δίνεται η εντολή για το κλείσιμο και συγκεκριμένα την επαναφορά του cooler στην θερμοκρασία των 26 C° που είναι η τιμή για την εξοικονόμηση ενέργειας. Συγκεκριμένα ενεργοποιήθηκε ο κανόνας:

*energy\_r22(Room, Device, energy), switchOff\_local(Room, Device, energy), []):-  
isOn(Device, Room), savingMode(Room), device(Device, heater\_cooler, Room)*

Ο κανόνας αυτός αφορά την πολιτική τύπου “energy management” η οποία έχει προτεραιότητα έναντι της πολιτικής του χρήστη (user). Εφόσον έχει ενεργοποιηθεί η λειτουργία saving και η συγκεκριμένη αίθουσα βρίσκεται ανάμεσα στις ζώνες με την μεγαλύτερη κατανάλωση ενέργειας, τότε θα πρέπει να οριστεί η θερμοκρασία του cooler στους 26 C°. Αντίστοιχα θα συνέβαινε αν η συσκευή ήταν heater.

### Σενάριο 7°

Το δωμάτιο που επιλέγουμε είναι τύπου “γραφείο” (office), ρυθμίζουμε την λειτουργία σε ψύξη (cool). Οι συσκευές heater, alarm είναι απενεργοποιημένες, ενώ ο cooler είναι ενεργοποιημένος. Ρυθμίζουμε την επιθυμητή θερμοκρασία (Target Temperature) σε χαμηλότερο σημείο από την υπάρχουσα (Current Temperature). Ο αισθητήρας κίνησης έχει καταγράψει παρουσία στον χώρο. Η ώρα είναι η υπάρχουσα, και στο πεδίο του energy policy βάζουμε την τιμή Critical και το thermal zone level στο επίπεδο 12. Το critical mode πρόκειται για το δεύτερο επίπεδο στην λειτουργία εξοικονόμησης ενέργειας. Σε αυτό το επίπεδο μπαίνουν οι ζώνες στην περίπτωση που υπάρχει μεγάλη κατανάλωση ενέργειας στο κτίριο και ο υπεύθυνος επιθυμεί να μειώσει με τον πιο δραστικό τρόπο την κατανάλωση αυτή. Ενεργοποιώντας αυτή την πολιτική, απενεργοποιούνται αυτόματα όλες οι συσκευές (heater, cooler) που βρίσκονται μέσα σε αυτές τις ζώνες. Το γραφείο αυτό ανήκει στην θερμικής ζώνη 20, η οποία βρίσκεται στην πρώτη θέση της κατάταξης των ζωνών με βάση την κατανάλωση τους. Εφόσον δηλαδή η τιμή που ορίζουμε στο Thermal zone level είναι μεγαλύτερη από την θέση κατάταξης που είναι το γραφείο, τότε μπαίνει στο critical mode. (Thermal zone level> Thermal zone ranking).

Rooms

Action

Cooler ☒

Heater ☐

Alarm ☐

Current Temperature

Target Temperature

Motion ☒

Current CO2

Current Time

Energy policy

Thermal zone level

## Results

Switch On cooler: false  
 Do not switch On cooler: false  
 Switch Off cooler: true:According to the energy managment policy,if the device (cooler,heater) is On and critical mode in On, then switch Off the device - ['E=[energy\_r\_crmo1(space\_110, cooler2, energy), r4(space\_110, cooler2, energy)]:']  
 Do not switch Off cooler: false  
 Switch On alarm: false  
 Do not switch On alarm: false  
 Switch Off alarm: false  
 Do not switch Off alarm: false

### EIKONA 22. Σενάριο 7

Εκτελώντας το πρόγραμμα παρατηρούμε ότι δίνεται η εντολή για το κλείσιμο του cooler. Συγκεκριμένα ενεργοποιήθηκε ο κανόνας:

*energy\_r\_crmo1(Room, Device, energy), switchOff\_local(Room, Device, energy), []):-criticalMode(Room), isOn(Device, Room), device(Device, heater\_cooler, Room)*

Ο κανόνας αυτός αφορά την πολιτική τύπου “energy management” η οποία έχει προτεραιότητα έναντι της πολιτική του χρήστη (user). Εφόσον έχει ενεργοποιηθεί η λειτουργία critical και η συγκεκριμένη αίθουσα βρίσκεται ανάμεσα στις ζώνες με την μεγαλύτερη κατανάλωση ενέργειας, τότε θα πρέπει να απενεργοποιηθεί ο cooler που βρίσκεται στον χώρο. Αντίστοιχα θα συνέβαινε αν η συσκευή ήταν heater.

### Σενάριο 8°

Επιλέγουμε το δωμάτιο που είναι τύπου αμφιθέατρο (auditorium), ρυθμίζουμε την λειτουργία σε ψύξη (cool). Όλες οι συσκευές cooler, heater, alarm είναι απενεργοποιημένες. Ρυθμίζουμε την επιθυμητή θερμοκρασία (Target Temperature) σε σημείο χαμηλότερο από την υπάρχουσα (Current Temperature). Ο αισθητήρας κίνησης έχει καταγράψει παρουσία στον χώρο. Το energy policy είναι απενεργοποιημένο στην λειτουργία none, όπως και το thermal zone level. Την ώρα σε αυτό το

παράδειγμα την ορίζουμε ως 11:52, που είναι εντός του χρονοδιαγράμματος (schedule) βάσει του οποίου λειτουργεί το αμφιθέατρο. Συγκεκριμένα έχουμε ορίσει τις ώρες λειτουργίας αυτού βάσει των μαθημάτων που γίνονται, 9:00 με 18:00.

**Rooms**

**Action**

**Cooler** ☐

**Heater** ☐

**Alarm** ☐

**Current Temperature**

**Target Temperature**

**Motion** ☒

**Current CO2**

**Current Time**

**Energy policy**

**Thermal zone level**

## Results

Switch On cooler: true:According to the user policy, if the device (cooler, heater) is Off and the temperature is out of the bounds, then switch On the device - ['E=[user\_r11(space\_121, cooler1, user), r1(space\_121, cooler1, user)]']  
Do not switch On cooler: false  
Switch Off cooler: false  
Do not switch Off cooler: false  
Switch On alarm: false  
Do not switch On alarm: false  
Switch Off alarm: false  
Do not switch Off alarm: false

EIKONA 23. Σενάριο 8

Εκτελώντας το πρόγραμμα παρατηρούμε ότι δίνεται η εντολή για το άνοιγμα του cooler. Συγκεκριμένα ενεργοποιήθηκε ο κανόνας:

*user\_r11(Room, Device, user), switchOn\_local(Room, Device, user), []):-isOff(Device, Room), tempOutOfBounds(Room), device(Device, heater\_cooler, Room)*

Αυτός ο κανόνας ο οποίος αφορά την πολιτική του χρήστη (user) και οι προϋποθέσεις ενεργοποίησης του ήταν: η θερμοκρασία να είναι εκτός των ορίων που έχει θέσει ο χρήστης, το σύστημα ψύξης είναι απενεργοποιημένο ενώ ταυτόχρονα υπάρχει παρουσία ανθρώπων στο χώρο.

### Σενάριο 9°

Επιλέγουμε το δωμάτιο που είναι τύπου αμφιθέατρο (auditorium), ρυθμίζουμε την λειτουργία σε ψύξη (cool). Οι συσκευές heater, alarm είναι απενεργοποιημένες, ενώ ο cooler είναι ενεργοποιημένος. Ρυθμίζουμε την επιθυμητή θερμοκρασία (Target Temperature) σε σημείο χαμηλότερο από την υπάρχουσα (Current Temperature). Ο αισθητήρας κίνησης έχει καταγράψει

παρουσία στον χώρο. Το energy policy είναι απενεργοποιημένο στην λειτουργία none, όπως και το thermal zone level. Η ώρα σε αυτό το παράδειγμα την ορίζουμε ως 20:52, που είναι εκτός του χρονοδιαγράμματος (schedule) βάσει του οποίου λειτουργεί το αμφιθέατρο.

**Rooms** Auditorium ▼

**Action** Cool ▼

**Cooler** ☒

**Heater** ☐

**Alarm** ☐

**Current Temperature** 23

**Target Temperature** 20

**Motion** ☒

**Current CO2** 200

**Current Time** 08:52 PM 🕒

**Energy policy** None ▼

**Thermal zone level** 1 ▼

Prove

## Results

Switch On cooler: false  
Do not switch On cooler: false  
Switch Off cooler: true:According to the user policy,if the device (cooler,heater) is On, then switch Off the device - ['!E=[nott(user\_c22(space\_121, cooler1, user)), user\_p22(space\_121, cooler1, user), user\_c22(space\_121, cooler1, user), user\_r22(space\_121, cooler1, user), r4(space\_121, cooler1, user)]]']  
Do not switch Off cooler: false  
Switch On alarm: false  
Do not switch On alarm: false  
Switch Off alarm: false  
Do not switch Off alarm: false

### EIKONA 24. Σενάριο 9

Παρατηρούμε σε αυτό το σενάριο ότι η εντολή που δίνεται είναι για απενεργοποίηση του cooler παρόλο που ο αισθητήρας καταγράφει παρουσία στον χώρο. Αυτό συμβαίνει καθώς στο συγκεκριμένο δωμάτιο, αμφιθέατρο, έχουμε δώσει μεγαλύτερη προτεραιότητα στο χρονοδιάγραμμα βάσει του οποίου λειτουργεί σε σχέση με τον αισθητήρα κίνησης.

### Σενάριο 10°

Σε αυτό το σενάριο επιλέγουμε το δωμάτιο τύπου computer room και ρυθμίζουμε την λειτουργία σε ψύξη (cool). Οι συσκευές heater, alarm είναι απενεργοποιημένες, ενώ ο cooler είναι ενεργοποιημένος. Ρυθμίζουμε την επιθυμητή θερμοκρασία (Target Temperature) σε σημείο χαμηλότερο από την υπάρχουσα (Current Temperature). Είναι σημαντικό να τονίσουμε ότι τα δωμάτια αυτού του τύπου έχουν ανάγκη διατήρησης σταθερής θερμοκρασία καθ' όλη την διάρκεια του έτους λόγω των αναγκών των υπολογιστών. Επίσης ο αισθητήρας κίνησης δεν έχει καταγράψει



παρουσία στον χώρο. Το energy policy είναι απενεργοποιημένο στην λειτουργία none, όπως και το thermal zone level.

**Rooms** 
  
**Action** 
  
**Cooler** ☒
  
**Heater** ☐
  
**Alarm** ☐
  
**Current Temperature** 
  
**Target Temperature** 
  
**Motion** ☐
  
**Current CO2** 
  
**Current Time** 
  
**Energy policy** 
  
**Thermal zone level**

## Results

Switch On cooler: false  
 Do not switch On cooler: false  
 Switch Off cooler: false  
 Do not switch Off cooler: true:According to the user policy,if the device (cooler,heater) is On, the temperature is out of bounds,then don't switch On the device - ['!E=[nott(c111(user, space\_115, cooler3, energy)), p\_pe\_en34(user, space\_115, cooler3, energy), c111(user, space\_115, cooler3, energy), user\_r21(space\_115, cooler3, user), r3(space\_115, cooler3, user)]]']  
 Switch On alarm: false  
 Do not switch On alarm: false  
 Switch Off alarm: false  
 Do not switch Off alarm: false

EIKONA 25. Σενάριο 10

Παρατηρούμε σε αυτό το σενάριο ότι η εντολή που δίνεται είναι μην γίνει απενεργοποίηση του cooler παρόλο που ο αισθητήρας κίνησης δεν καταγράφει παρουσία στον χώρο. Οι κανόνες που ενεργοποιήθηκαν είναι οι εξής:

*Not(c111(user, Room, Device, energy), prefer(p\_pe\_en12(user, Room, Device, energy), p\_pe\_en21(energy, Room, Device, user))*

Αυτό συμβαίνει καθώς στο συγκεκριμένο δωμάτιο, computer room, η πολιτική του user υπερισχύει από την πολιτική του energy management. Δηλαδή ακόμη και στην περίπτωση που εξετάζουμε χωρίς την παρουσία ατόμων στον χώρο που κανονικά θα απενεργοποιείτο ο cooler, στο συγκεκριμένο δωμάτιο παραμένει ανοιχτός. Ο κανόνας που μας το δείχνει αυτό είναι ο:

*p\_pe\_en21(user, Room, Device, energy), prefer(r2(Room, Device, user), r1(Room, Device, energy))*

## Κεφάλαιο 6°

### Αποτελέσματα εργασίας

Η εργασία, όπως παρουσιάστηκε στα προηγούμενα κεφάλαια, αποτελεί μια ολοκληρωμένη λύση για την λήψη και εφαρμογή αποφάσεων σε έξυπνα ενεργειακά κτίρια. Το κτίριο K2 της σχολής μηχανικών Περιβάλλοντος του Πολυτεχνείου Κρήτης αποτέλεσε το ιδανικό παράδειγμα μελέτης περίπτωσης για την εφαρμογή που αναπτύχθηκε. Ο λόγος είναι οι μεγάλες ανάγκες σε ενέργεια που απαιτεί το κτίριο καθώς και τα πολλά, διαφορετικά επίπεδα διαχείρισης του κτιρίου που λειτουργούν με διαφορετικούς κανόνες. Για παράδειγμα οι αίθουσες διδασκαλίας, οι οποίες λειτουργούν βάσει συγκεκριμένου προγράμματος, και τα γραφεία, τα οποία παρουσιάζουν μεγάλη μεταβλητότητα στις ώρες λειτουργίας τους.

Η εφαρμογή που υλοποιήθηκε χαρακτηρίζεται από την απλότητα αλλά, ταυτόχρονα, και την επεκτασιμότητα της.

Στο κομμάτι της απλότητας είναι σημαντικό να τονίσουμε την ευκολία που παρέχει το εργαλείο Gorgias Cloud, όπου είναι βασισμένο στη γλώσσα προγραμματισμού Prolog. Η γλώσσα αυτή έχει ως βασικό χαρακτηριστικό την συγγραφή κώδικα σε φυσική γλώσσα, κάτι που οδηγεί στην εύκολη εκμάθηση της και συνεπώς στην δημιουργία εντολών για τον Gorgia.

Αναφορικά με την επεκτασιμότητα, στην συγκεκριμένη εφαρμογή χρησιμοποιήθηκαν τρεις διαφορετικές πολιτικές: του χρήστη (user), του ενεργειακού διαχειριστή (energy management) και της πολιτικής έκτακτης ανάγκης (emergency). Αυτές οι τρεις διαφορετικές πολιτικές συνδυάστηκαν και προτεραιοποιήθηκαν βάσει των αναγκών της παρούσας εργασίας. Η προτίμηση μιας πολιτικής έναντι μιας άλλης έγινε με την χρήση ενός συνόλου απλών κανόνων που φαίνεται παρακάτω:

*rule(p1(emergency), prefer(r3(emergency), r1(personal)), []).*

*rule(p2(emergency), prefer(r3(emergency), r2(management)), []).*

*rule(p3(management), prefer(r2(management), r1(personal)), []).*

*rule(p4(personal), prefer(r1(personal), r2(management)), []):-  
device(Device,cooler,Room),computerRoom(Room).*

*rule(c1(personal), prefer(p4(personal), p3(management)), []).*

Συνεπώς, διαπιστώνουμε ότι μπορούν να δημιουργηθούν, με εύκολο τρόπο πολιτικές, οι οποίες καλύπτουν τις εκάστοτε ανάγκες. Επιπλέον, οι πολιτικές αυτές μπορούν να προτεραιοποιηθούν με εξίσου εύκολο τρόπο, ώστε να αποφευχθούν συγκρούσεις. Η ιεραρχική επιχειρηματολογία, λοιπόν, αποτελεί το κατάλληλο εργαλείο για την επίλυση συγκρούσεων και για λήψη αποφάσεων με

εύκολο και ξεκάθαρο τρόπο καθώς, μέσω του γραφικού περιβάλλοντος που αναπτύχθηκε, μπορεί ο χρήστης να διαπιστώσει μόνος του ποιος κανόνας ενεργοποιήθηκε κάθε φορά.

Συνοψίζοντας, τα σημαντικότερα πορίσματα που απορρέουν είναι:

1. Εξοικονόμηση ενέργειας από την ορθολογική χρήση μέσων όπως η θέρμανση/ψύξη και επομένως, αναβάθμιση της ποιότητας ζωής,
2. Αυτόματη λήψη αποφάσεων, ακόμη και χωρίς τον ανθρώπινο παράγοντα,

Τέλος,

3. Αν χρησιμοποιηθεί η στατιστική παρακολούθηση και ανάλυση των δεδομένων, από τα υποπροϊόντα, μπορούν να παραχθούν αξιόλογα αποτελέσματα, με μεγάλο εύρος εφαρμοσιμότητας, όπως, για παράδειγμα, σε κλιματικές εφαρμογές για την εκτίμηση του κόστους-οφέλους συγκεκριμένων ενεργειών. Π.χ. Συμπεράσματα από τη χρήση μόνωσης ή σχετικά με τη χρήση κάποιων ηλεκτρικών συσκευών ή όσον αφορά την κατανάλωση νερού.

Η “ιεραρχική επιχειρηματολογία”, η “υποστήριξη λήψης αποφάσεων”, αποτελούν τις λέξεις-κλειδιά που καθιστούν το σύστημα που έχει αναπτυχθεί (και που μπορεί δυναμικά να εξελιχθεί περαιτέρω), διαθέσιμο για πλήθος μελλοντικών εφαρμογών, σε διάφορους χώρους της επιστήμης. Ανθρωποκεντρική εφαρμογή, με στόχο την εξυπηρέτηση του ανθρώπου με κάθε σύγχρονο τρόπο, έχει το βλέμμα στο μέλλον, καθώς βρίσκεται στην αιχμή της τεχνολογίας στην τεχνητή νοημοσύνη και στην λήψη αποφάσεων που είναι στηριγμένη στην Λογική.

## Κεφάλαιο 7°

### Σύνοψη και μελλοντική εργασία

Σκοπός αυτής της εργασίας ήταν η εφαρμογή του εργαλείου Gorgias στην λήψη αποφάσεων για θέματα ενέργειας σε “έξυπνα” κτίρια. Συνδυαστικά με το πρόγραμμα Open studio όπου χρησιμοποιήθηκε η μοντελοποίηση του κτιρίου K2 στο Πολυτεχνείο Κρήτης, λαμβάνοντας άκρως χρήσιμα δεδομένα. Αξιοποιώντας τις ιδιότητες του κτιρίου σαν σύνολο αλλά και των μεμονωμένων αιθουσών που βρίσκονται σε αυτό προέκυψαν σημαντικά στοιχεία για την ενεργειακή απόδοση και την κατανάλωση ενέργειας σε αυτό.

Στα προηγούμενα κεφάλαια παρουσιάστηκαν τα αποτελέσματα της εργασίας τα οποία διαθέτουν περιθώριο για περαιτέρω ανάπτυξη.

Όσον αφορά την δομή, η εργασία αναπτύχθηκε σε τέσσερα βασικά τμήματα. Η συλλογή μετρήσεων από τους αισθητήρες η οποία έγινε μέσω του Rest Api που αναπτύχθηκε σε python. Η χρήση και ανάλυση των δεδομένων που προέκυψαν μέσω της μοντελοποίησης του κτιρίου K2, της Σχολής Μηχανικών Περιβάλλοντος από το πρόγραμμα Open Studio. Η δημιουργία Rest Api του προγράμματος Gorgias, ώστε να γίνεται η εισαγωγή των παραμέτρων για την λήψη της απόφασης. Το γραφικό περιβάλλον διεπαφής του χρήστη (GUI) μέσω του οποίου παρουσιάζονται τα δεδομένα από τους αισθητήρες, σε όποιους χώρους υπάρχουν, ή μπορούν να εισαχθούν από τον χρήστη μια σειρά από στοιχεία ώστε να γίνει η λήψη της απόφασης και η παρουσίαση της στην οθόνη. Αυτά τα τέσσερα διαφορετικά κομμάτια συνδυάστηκαν μεταξύ τους ώστε μπορούν να λειτουργούν ως ένα σύστημα λήψης αποφάσεων.

Οι προεκτάσεις που θα μπορούσαν να γίνουν είναι αναφορικά τόσο με το γραφικό περιβάλλον όσο και με την συλλογή περισσότερων δεδομένων για την ορθότερη λήψη αποφάσεων.

Στο κομμάτι του γραφικού περιβάλλοντος θα μπορούσε να χρησιμοποιηθεί κάποιο framework όπως η Bootstrap, η οποία περιέχει ένα σύνολο εργαλείων για μορφοποίηση του περιβάλλοντος εργασίας. Με αυτό τον τρόπο η οθόνη εισαγωγής των δεδομένων αλλά και των αποτελεσμάτων θα είναι αισθητικά πιο όμορφη και συνεπώς πιο εύχρηστη από τον χρήστη.

Στο κομμάτι των δεδομένων, είναι δυνατόν να χρησιμοποιηθούν περισσότεροι αισθητήρες οι οποίοι να καλύπτουν συνολικά τις ανάγκες όχι μόνο του κτιρίου K2 αλλά του συνόλου των κτιρίων των πέντε σχολών του Πολυτεχνείου. Ειδικά σε αίθουσες διδασκαλίας-αμφιθέατρο που λόγω του μεγάλο όγκου τους παρουσιάζουν αυξημένες ανάγκες σε ενέργεια για την ψύξη και την θέρμανση τους. Οι αισθητήρες αυτοί μπορούν να επεκταθούν πέραν από αριθμό και σε δυνατότητες σε σχέση με αυτούς που διαθέτουμε, που μετρούν: θερμοκρασία, υγρασία, διοξείδιο του άνθρακα, παρουσία στο χώρο και κατανάλωση ενέργειας. Με την προσθήκη κατάλληλων αισθητήρων μέτρησης κατανάλωσης νερού στις τουαλέτες, στις κουζίνες αλλά και στο σύστημα ποτίσματος της Πολυτεχνειούπολης θα μεγάλωνε σημαντικά ο αριθμός των δεδομένων που διαθέτει το Πολυτεχνείο για την λήψη αποφάσεων. Τέτοια δεδομένα θα είχαν πολύ μεγάλο ενδιαφέρον να

ενσωματωθούν στο σύστημα ώστε οι αποφάσεις που λαμβάνονται να αφορούν ένα μεγαλύτερο εύρος και μεγαλύτερη αποτελεσματικότητα ως προς την μείωση των καταναλώσεων. Επιπλέον πέραν των αισθητήρων θα μπορούσαν να υπάρχουν στο σύστημα και ενεργοποιητές οι οποίοι θα εκτελούν την απόφαση που έχει ληφθεί από το σύστημα. Με αυτό τον τρόπο ο κλιματισμός/θέρμανση και ο συναγερμός θα ενεργοποιούνταν ή απενεργοποιούνταν αυτόματα.

Τέλος το Πολυτεχνείο Κρήτης έχει αναπτύξει την φιλοσοφία του Πράσινου Πολυτεχνείου ως αναγκαιότητα λόγω της οικονομικής κρίσης αλλά και ταυτόχρονα θέτοντας ως στόχο την βιώσιμη ανάπτυξη. Συνεπώς μέσα στο Στρατηγικό Σχέδιο βιώσιμης Ανάπτυξης το οποίο βρίσκεται σε φάση διαβούλευσης θα ήταν δυνατόν να ενταχθεί και το παρόν σύστημα λήψης αποφάσεων. Αρχικά έχοντας ένα ρόλο συμβουλευτικό στον ενεργειακό διαχειριστή των κτιρίων αλλά με στόχο στο άμεσο μέλλον να αποτελεί ένα έξυπνο και αυτόνομο σύστημα λήψης αποφάσεων σε πραγματικό χρόνο. Έτσι με αυτό τον τρόπο θα υπάρξουν μετρήσιμα αποτελέσματα, όπως μείωση κατανάλωσης ενέργειας, νερού και εκπομπών διοξειδίου του άνθρακα αλλά και ταυτόχρονα μείωση του κόστους.

## Βιβλιογραφία

- [1] T. Malche and P. Maheshwary, "Internet of Things (IoT) for building smart home system," in *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, Oct. 2017, pp. 65–70, doi: 10.1109/I-SMAC.2017.8058258.
- [2] D. Kolokotsa *et al.*, "Development of a web based energy management system for University Campuses: The CAMP-IT platform," *Energy Build.*, vol. 123, pp. 119–135, Jul. 2016, doi: 10.1016/j.enbuild.2016.04.038.
- [3] K. Gobakis, A. Mavrigiannaki, K. Kalaitzakis, and D. D. Kolokotsa, "Design and development of a Web based GIS platform for zero energy settlements monitoring," in *Energy Procedia*, Oct. 2017, vol. 134, pp. 48–60, doi: 10.1016/j.egypro.2017.09.598.
- [4] T. G. Stavropoulos *et al.*, "Rule-based approaches for energy savings in an ambient intelligence environment," *Pervasive Mob. Comput.*, vol. 19, pp. 1–23, May 2015, doi: 10.1016/j.pmcj.2014.05.001.
- [5] N. I. Spanoudakis, A. C. Kakas, and P. Moraitis, "Applications of argumentation: The SoDA methodology," in *22nd European Conference on Artificial Intelligence (ECAI)*, 2016, vol. 285, pp. 1722–1723, doi: 10.3233/978-1-61499-672-9-1722.
- [6] A. C. Kakas, P. Moraitis, and N. I. Spanoudakis, "GORGIAS: Applying argumentation," *Argument Comput.*, vol. 10, no. 1, pp. 55–81, Jan. 2019, doi: 10.3233/AAC-181006.
- [7] N. Bassiliades, N. I. Spanoudakis, and A. C. Kakas, "Towards Multipolicy Argumentation," in *Proceedings of the 10th Hellenic Conference on Artificial Intelligence - SETN '18*, 2018, p. 10, Accessed: Dec. 01, 2020. [Online]. Available: <https://doi.org/10.1145/3200947.3201032>.
- [8] Γληγόρης Γεώργιος, "Επιχειρηματολογία και Συλλογισμός στο Υπολογιστικό Νέφος," Technical University of Crete, 2020.
- [9] KNX Association, "KNX System Specifications," vol. 2.1, p. 8, 2013.
- [10] "SOAP vs. REST: A Look at Two Different API Styles | Upwork." <https://www.upwork.com/resources/soap-vs-rest-a-look-at-two-different-api-styles> (accessed Jul. 19, 2021).
- [11] "API Documentation & Design Tools for Teams | Swagger." <https://swagger.io/> (accessed Jul. 19, 2021).
- [12] "PyCharm: the Python IDE for Professional Developers by JetBrains." <https://www.jetbrains.com/pycharm/> (accessed Jul. 19, 2021).
- [13] "The Web framework for perfectionists with deadlines | Django." <https://www.djangoproject.com/> (accessed Jul. 19, 2021).

- [14] KNX Association, “ETS5: one tool for ALL media.”
- [15] “Γοργίαις - Βικιπαίδεια.” <https://el.wikipedia.org/wiki/Γοργίαις> (accessed Jul. 19, 2021).
- [16] Weinzierl Engineering GmbH, “BAOS REST-Services API v3.3 -,” pp. 1–51, [Online]. Available: [https://weinzierl.de/images/download/documents/baos/KNX\\_IP\\_BAOS\\_RESTServices.pdf](https://weinzierl.de/images/download/documents/baos/KNX_IP_BAOS_RESTServices.pdf).
- [17] “Πολυτεχνείο Κρήτης: Αρχή.” <https://www.tuc.gr/index.php?id=4992> (accessed Jul. 19, 2021).
- [18] Weinzierl Engineering GmbH, “Knx Ip Baos 777-Datasheet,” vol. 777, pp. 1–17, 2018, [Online]. Available: <https://www.weinzierl.de/images/download/products/777/Weinzierl-777-KNX-IP-BAOS-5193-Datasheet-EN.pdf>.
- [19] ABB Home and Building Automation, “Energy Module , MDRC Energy Module , MDRC,” pp. 1–6.
- [20] ABB Home and Building Automation, “PM2.5 Air Quality Sensor,” pp. 2–17, 2020, [Online]. Available: <https://adafru.it/CiF>.
- [21] ABB Home and Building Automation, “P R O D U C T S E L E C T I O N G U I D E ABB i-bus ® KNX Movement and Presence Detectors ABB i-bus ® KNX Movement and Presence Detectors,” pp. 1–2, 2017.

# Appendix

## KNX BAOS REST API

```
import requests
import json
from collections import namedtuple
import sched, time
import os
import os.path
import csv
from datetime import datetime

clear = lambda: os.system('cls')
clear()

headers = {'Content-type': 'application/json, text/plain, /*'}
s.headers.update(headers);
s.headers.update({
    'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36', })
headers = {'Accept-Language': 'el-GR,el;q=0.9,en;q=0.8'}
s.headers.update(headers);

url = 'http://147.27.33.130/rest/login'
pload = {'username': '****', 'password': '****'}
x = s.post(url, json=pload, headers=headers)

TOKEN = x.text
HEADERS = {'Cookie': 'user=%22' + TOKEN + '%22'}
s.headers.update(HEADERS)

URL = 'http://147.27.33.130/rest/device'
r = s.get(URL)
x = json.loads(r.text, object_hook=lambda d: namedtuple('X', d.keys())(*d.values()))
DevName = "Device {}, Version {}".format(x.device.name, x.device.build_version)
Devfilename = "{}_{}.csv".format(x.device.name, x.device.build_version)

URL = 'http://147.27.33.130/rest/serveritems/values?count=1&format=true&start=39'
r = s.get(URL)
x = json.loads(r.text, object_hook=lambda d: namedtuple('X', d.keys())(*d.values()))
DevCount = x.serveritems_values[0].value

URL = 'http://147.27.33.130/rest/datapoints/descriptions?count=25&page=1'
r = s.get(URL)
DevNames = json.loads(r.text, object_hook=lambda d: namedtuple('X', d.keys())(*d.values()))

def do_something(sc):

    clear()

    URL = 'http://147.27.33.130/rest/datapoints/values?start=1&end=25'
```



```

52     r = s.get(URL)
53     x = json.loads(r.text, object_hook=lambda d: namedtuple('X', d.keys())(*d.values()))
54
55     print(DevName)
56     print('-----')
57
58     for i in range(DevCount):
59         txt3 = "{:<12} Id {} {} is_update {} is_valid {} value {} time {}".format(
60             DevNames.datapoints_descriptions[i].name, x.datapoints_values[i].id,
61             x.datapoints_values[i].Format,
62             x.datapoints_values[i].state.is_update, x.datapoints_values[i].state.is_valid,
63             x.datapoints_values[i].value,
64             datetime.now())
65         print(txt3)
66
67
68         txt3 = "{,0,0,0,0,0,0}".format(DevNames.datapoints_descriptions[i].name,
69             x.datapoints_values[i].id,
70             x.datapoints_values[i].Format, x.datapoints_values[i].state.is_update,
71             x.datapoints_values[i].state.is_valid, x.datapoints_values[i].value)
72
73         with open(Devfilename, 'a+', newline="") as file:
74             writer = csv.writer(file)
75             writer.writerow([txt3])
76
77         ss.enter(5, 1, do_something, (sc,))
78
79     if os.path.isfile(Devfilename) == False:
80         with open(Devfilename, 'a+', newline="") as file:
81             writer = csv.writer(file)
82             writer.writerow(["Name", "ID", "Name", "is_update", "is_valid", "value"])
83
84     ss.enter(1, 1, do_something, (s,))
85     ss.run()

```

## 86 Gorgias REST API

```

87 from django.http import HttpResponse, JsonResponse
88 from django.shortcuts import render
89 from django.views.decorators.csrf import csrf_exempt
90
91
92 def makeResult(api_response):
93     print(api_response)
94     naturalLanguage = {
95         "user_r11": "According to the user policy, if the device (cooler, heater) is Off and the
96         temperature is out of the bounds, then switch On the device" ,
97         "user_r12": "According to the user policy,if the device (cooler,heater) is Off, then
98         don't switch On the device",
99         "user_p12": "According to the user policy, if the current time is not within the time
100        schedule of that room, prefer temperature is out of the bounds that user has defined",
101         "user_r21": "According to the user policy,if the device (cooler,heater) is On, the

```

```

102 temperature is out of bounds,then don't switch On the device",
103     "user_r22": "According to the user policy,if the device (cooler,heater) is On, then
104     switch Off the device",
105     "energy_r_mose1": "According to the energy managment policy,if the device
106     (cooler,heater) is On and there in no motion in the room is out of bounds,then switch Off
107     the device",
108     "energy_r_mose2": "According to the energy managment policy,if the device
109     (cooler,heater) is Off and there in no motion in the room is out of bounds,then don't switch
110     On the device",
111     "energy_r11": "According to the energy managment policy,if the device
112     (cooler,heater) is Off, saving mode in On and there the temperature is out of bounds, then
113     switch On the device",
114     "energy_r12": "According to the energy managment policy,if the device
115     (cooler,heater) is Off and saving mode in On, then don't switch On the device",
116     "energy_r21": "According to the energy managment policy,if the device
117     (cooler,heater) is On,saving mode in On and the temperature is out of bounds, then don't
118     switch Off the device",
119     "energy_r22": "According to the energy managment policy,if the device
120     (cooler,heater) is On and saving mode in On, then switch Off the device",
121     "energy_r_crmo1": "According to the energy managment policy,if the device
122     (cooler,heater) is On and critical mode in On, then switch Off the device",
123     "energy_r_crmo2": "According to the energy managment policy,if the device
124     (cooler,heater) is Ogf and critical mode in On, then don't switch On the device",
125     "emergency_r_em1": "According to the emergency policy,if the device (cooler,heater)
126     is On and there is an alert on the room, then switch Off the device",
127     "emergency_r_em4": "According to the emergency policy,if the alrarm is On and
128     there is an alert on the room, then don't switch Off the alarm",
129     "emergency_r_em2": "According to the emergency policy,if the device (cooler,heater)
130     is Off and there is an alert on the room, then don't switch On the device",
131     "emergency_r_em3": "According to the emergency policy,if the alrarm is Off and
132     there is an alert on the room, then switch On the alarm",
133     "p_pe_en12": "user's policy is preferred over the energy manager's policy because
134     the room is the computer room and the device the cooler"
135 }
136
137
138 api_response = str(api_response)
139 print(api_response)
140 if api_response == '["false"]':
141     return 'false\n'
142
143 for case in naturalLanguage.keys():
144     if api_response.find(case) >= 0:
145         print('case', case)
146         return 'true:' + naturalLanguage[case] + ' - ' + api_response + '\n'
147
148 return 'true: ' + api_response + '\n'
149
150
151 @csrf_exempt
152 def gorgias(request):
153     import json
154     data = json.loads(request.body)

```

```

155     print(data)
156
157     import swagger_client
158     from swagger_client.rest import ApiException
159
160     configuration = swagger_client.Configuration()
161     configuration.username = '****'
162     configuration.password = '****'
163     configuration.host = "http://aiasvm1.amcl.tuc.gr:8085"
164
165
166     api_instance =
167     swagger_client.OperationsControllerApi(swagger_client.ApiClient(configuration))
168     api_instance2 = swagger_client.PrologControllerApi(swagger_client.ApiClient(configuration))
169
170     project = 'final'
171
172     api_response = api_instance2.consult_file_using_post('combinedPolicy.pl', project)
173     print(api_response)
174
175     roomAssignment = {
176
177         "room1": "space_121",
178         "room2": "space_110",
179         "room3": "space_115",
180
181     }
182
183     coolerAssignment = {
184         "room1": "cooler1",
185         "room2": "cooler2",
186         "room3": "cooler3",
187     }
188
189     heaterAssignment = {
190         "room1": "heater1",
191         "room2": "heater2",
192         "room3": "heater3",
193     }
194
195     alarmAssignment = {
196         "room1": "alarm1",
197         "room2": "alarm2",
198         "room3": "alarm3",
199     }
200
201     sensorAssignment = {
202         "room1": "sensor1",
203         "room2": "sensor2",
204         "room3": "sensor3",
205     }
206     result = ""
207     try:

```

```

208     A1 = {"query": "asserta(hasTemp({},{})).".format(roomAssignment[data["rooms"]],
209 int(float(data["currentTemp"]))), "result_size": 1, "time": 1000}
210     api_response = api_instance2.prolog_command_using_post(A1)
211     print(A1, api_response)
212     A2 = {"query": "asserta(limitByUser({},{})).".format(roomAssignment[data["rooms"]],
213 int(float(data["targetTemp"])), data["action"]), "result_size": 1, "time": 1000}
214     api_response = api_instance2.prolog_command_using_post(A2)
215     print(A2, api_response)
216     A3 = {"query": "asserta(reading({},{})).".format(sensorAssignment[data["rooms"]], 'true' if
217 data["motionState"] else 'false'), "result_size": 1, "time": 1000}
218     api_response = api_instance2.prolog_command_using_post(A3)
219     print(A3, api_response)
220     A4 = {"query": "asserta({},{},{})).".format('isOn' if data['coolerState'] else
221 'isOff', coolerAssignment[data['rooms']], roomAssignment[data['rooms']], "result_size": 1,
222 "time": 1000}
223     api_response = api_instance2.prolog_command_using_post(A4)
224     print(A4, api_response)
225     A5 = {"query": "asserta({},{},{})).".format('isOn' if data['heaterState'] else
226 'isOff', heaterAssignment[data['rooms']], roomAssignment[data['rooms']], "result_size": 1,
227 "time": 1000}
228     api_response = api_instance2.prolog_command_using_post(A5)
229     print(A5, api_response)
230     A6 = {"query": "asserta({},{},{})).".format('isOn' if data['alarmState'] else
231 'isOff', alarmAssignment[data['rooms']], roomAssignment[data['rooms']], "result_size": 1,
232 "time": 1000}
233     api_response = api_instance2.prolog_command_using_post(A6)
234     print(A6, api_response)
235     A7 = {"query": "asserta(limit({},{})).".format(data["thermalZoneLevel"],
236 data["energyPolicy"]), "result_size": 1, "time": 1000}
237     api_response = api_instance2.prolog_command_using_post(A7)
238     print(A7, api_response)
239     A8 = {"query": "asserta(current_time({})).".format(int(data["currentTime"].split(':')[0]) *
240 100), "result_size": 1, "time": 1000}
241     api_response = api_instance2.prolog_command_using_post(A8)
242     print(A8, api_response)
243     A9 = {"query": "asserta(co2Level({},{})).".format(roomAssignment[data["rooms"]],
244 int(float(data["currentCO2"]))), "result_size": 1, "time": 1000}
245     api_response = api_instance2.prolog_command_using_post(A9)
246     print(A9, api_response)
247
248     if data["action"] == 'cool':
249         P1 = {"query": "prove([switchOn({},{}),E)".format(roomAssignment[data["rooms"]],
250 coolerAssignment[data["rooms"]]), "result_size": 1, "time": 10000}
251         api_response = api_instance2.prove_using_post(P1)
252         print(P1, api_response)
253         result += 'Switch On cooler: ' + makeResult(api_response)
254         P12 = {"query":
255 "prove([neg(switchOn({},{})),E)".format(roomAssignment[data["rooms"]], coolerAssignment[dat
256 a["rooms"]]), "result_size": 1, "time": 10000}
257         api_response = api_instance2.prove_using_post(P12)
258         print(P12, api_response)
259         result += 'Do not switch On cooler: ' + makeResult(api_response)
260         P2 = {"query":

```

```

261 "prove([switchOff({},{}),E)".format(roomAssignment[data['rooms']],coolerAssignment[data['rooms']], "result_size": 1,"time": 10000}
262 api_response = api_instance2.prove_using_post(P2)
263 print(P2, api_response)
264 result += 'Switch Off cooler: ' + makeResult(api_response)
265 P22 = {"query":
266 "prove([neg(switchOff({},{})),E)".format(roomAssignment[data['rooms']],coolerAssignment[dat
267 a['rooms']], "result_size": 1, "time": 10000}
268 api_response = api_instance2.prove_using_post(P22)
269 print(P22, api_response)
270 result += 'Do not switch Off cooler: ' + makeResult(api_response)
271 else:
272 P3 = {"query":
273 "prove([switchOn({},{}),E)".format(roomAssignment[data['rooms']],heaterAssignment[data['rooms']], "result_size": 1,"time": 10000}
274 api_response = api_instance2.prove_using_post(P3)
275 print(P3, api_response)
276 result += 'Switch On heater: ' + makeResult(api_response)
277 P32 = {"query":
278 "prove([neg(switchOn({},{})),E)".format(roomAssignment[data['rooms']],heaterAssignment[dat
279 a['rooms']], "result_size": 1,"time": 10000}
280 api_response = api_instance2.prove_using_post(P32)
281 print(P32, api_response)
282 result += 'Do not switch On heater: ' + makeResult(api_response)
283 P4 = {"query": "prove([switchOff({},{}),E)".format(roomAssignment[data['rooms']],
284 heaterAssignment[data['rooms']], "result_size": 1,"time": 10000}
285 api_response = api_instance2.prove_using_post(P4)
286 print(P4, api_response)
287 result += 'Switch Off heater: ' + makeResult(api_response)
288 P42 = {"query":
289 "prove([neg(switchOff({},{})),E)".format(roomAssignment[data['rooms']],heaterAssignment[dat
290 a['rooms']], "result_size": 1,"time": 10000}
291 api_response = api_instance2.prove_using_post(P42)
292 print(P42, api_response)
293 result += 'Do not switch Off heater: ' + makeResult(api_response)
294 P5 = {"query": "prove([switchOn({},{}),E)".format(roomAssignment[data['rooms']],
295 alarmAssignment[data['rooms']], "result_size": 1, "time": 10000}
296 api_response = api_instance2.prove_using_post(P5)
297 print(P5, api_response)
298 result += 'Switch On alarm: ' + makeResult(api_response)
299 P52 = {"query": "prove([neg(switchOn({},{})),E)".format(roomAssignment[data['rooms']],
300 alarmAssignment[data['rooms']], "result_size": 1, "time": 10000}
301 api_response = api_instance2.prove_using_post(P52)
302 print(P52, api_response)
303 result += 'Do not switch On alarm: ' + makeResult(api_response)
304 P6 = {"query":
305 "prove([switchOff({},{}),E)".format(roomAssignment[data['rooms']],alarmAssignment[data['rooms']], "result_size": 1,"time": 10000}
306 api_response = api_instance2.prove_using_post(P6)
307 print(P6, api_response)
308 result += 'Switch Off alarm: ' + makeResult(api_response)
309 P62 = {"query":
310 "prove([neg(switchOff({},{})),E)".format(roomAssignment[data['rooms']],alarmAssignment[dat

```

```

314 a['rooms']], "result_size": 1, "time": 10000}
315     api_response = api_instance2.prove_using_post(P62)
316     print(P62, api_response)
317     result += 'Do not switch Off alarm: ' + makeResult(api_response)
318
319     R1 = {"query": "retract(hasTemp(X,Y)).", "result_size": 20, "time": 1000}
320     retract = api_instance2.prolog_command_using_post(R1)
321     print('R1', retract)
322     R2 = {"query": "retract(limitByUser(A,B,C)).", "result_size": 20, "time": 1000}
323     retract = api_instance2.prolog_command_using_post(R2)
324     print('R2', retract)
325     R3 = {"query": "retract(isOff(X,Y)).", "result_size": 20, "time": 1000}
326     retract = api_instance2.prolog_command_using_post(R3)
327     print('R3', retract)
328     R4 = {"query": "retract(isOn(X,Y)).", "result_size": 20, "time": 1000}
329     retract = api_instance2.prolog_command_using_post(R4)
330     print('R4', retract)
331     R5 = {"query": "retract(current_time(X)).", "result_size": 20, "time": 1000}
332     retract = api_instance2.prolog_command_using_post(R5)
333     print('R5', retract)
334     R6 = {"query": "retract(reading(X,Y)).", "result_size": 20, "time": 1000}
335     retract = api_instance2.prolog_command_using_post(R6)
336     print('R6', retract)
337     R7 = {"query": "retract(limit(X,Y)).", "result_size": 20, "time": 1000}
338     retract = api_instance2.prolog_command_using_post(R7)
339     print('R7', retract)
340     R8 = {"query": "retract(co2Level(X,Y)).", "result_size": 20, "time": 1000}
341     retract = api_instance2.prolog_command_using_post(R8)
342     print('R8', retract)
343
344     unload = api_instance2.unload_file_using_post('combinedPolicy.pl', project)
345     except ApiException as e:
346         print("Exception: %s\n" % e)
347         return HttpResponse(500)
348
349     print(result)
350     return JsonResponse(result, safe=False)
351
352
353 def index(request):
354     return render(request, 'home.html')

```

## 355 HTLM Form

```

356 <!DOCTYPE html>
357 <html lang="en">
358 <head>
359     <meta charset="UTF-8">
360     <title>Home</title>
361
362     {% load static %}
363     <style>

```

```

364     body {
365         padding: 15px;
366     }
367
368     form div, button {
369         margin: 10px;
370     }
371
372     form label {
373         font-size: large;
374         font-weight: bold;
375         margin-right: 5px;
376     }
377 </style>
378 </head>
379 <body>
380 <h1>Automated decision making tool</h1>
381 <form action="">
382     <div>
383         <label for="rooms">Rooms</label>
384         <select name="rooms" id="rooms" onchange="onSelectChange(this)">
385             <option value="room1">Auditorium</option>
386             <option value="room2">Office (Sensor)</option>
387             <option value="room3">Computer room</option>
388         </select>
389     </div>
390
391     <div>
392         <label for="action">Action</label>
393         <select name="action" id="action">
394             <option value="cool">Cool</option>
395             <option value="heat">Heat</option>
396         </select>
397     </div>
398
399     <div>
400         <label for="coolerState">Cooler</label>
401         <input id="coolerState" type="checkbox">
402     </div>
403
404     <div>
405         <label for="heaterState">Heater</label>
406         <input id="heaterState" type="checkbox">
407     </div>
408
409     <div>
410         <label for="alarmState">Alarm</label>
411         <input id="alarmState" type="checkbox">
412     </div>
413
414     <div>
415         <label for="currentTemp">Current Temperature</label>
416         <input id="currentTemp" type="number">

```



```

417     </div>
418
419     <div>
420         <label for="targetTemp">Target Temperature</label>
421         <input id="targetTemp" type="number">
422     </div>
423
424     <div>
425         <label for="motionState">Motion</label>
426         <input id="motionState" type="checkbox">
427     </div>
428
429     <div>
430         <label for="currentCO2">Current CO2</label>
431         <input id="currentCO2" type="number">
432     </div>
433
434     <div>
435         <label for="currentTime">Current Time</label>
436         <input id="currentTime" type="time">
437     </div>
438
439     <div>
440         <label for="energyPolicy">Energy policy</label>
441         <select name="energyPolicy" id="energyPolicy">
442             <option value="none">None</option>
443             <option value="saving">Saving</option>
444             <option value="critical">Critical</option>
445         </select>
446     </div>
447
448
449     <div>
450         <label for="thermalZoneLevel">Thermal zone level</label>
451         <select name="thermalZoneLevel" id="thermalZoneLevel">
452             <option value="1">1</option>
453             <option value="2">2</option>
454             <option value="3">3</option>
455             <option value="4">4</option>
456             <option value="5">5</option>
457             <option value="6">6</option>
458             <option value="7">7</option>
459             <option value="8">8</option>
460             <option value="9">9</option>
461             <option value="10">10</option>
462             <option value="11">11</option>
463             <option value="12">12</option>
464         </select>
465     </div>
466
467     <button type="button" id="proveBtn" onclick="prove()">Prove</button>
468 </form>
469

```



```

470 <h1 id="resultsHeader" hidden>Results</h1>
471 <p id="resultsParagraph" style="white-space: pre-line"></p>
472
473 </body>
474 <script>
475     const rooms = document.getElementById("rooms");
476     const action = document.getElementById("action");
477     const coolerState = document.getElementById("coolerState");
478     const heaterState = document.getElementById("heaterState");
479     const alarmState = document.getElementById("alarmState");
480     const currentTemp = document.getElementById("currentTemp");
481     const targetTemp = document.getElementById("targetTemp");
482     const motionState = document.getElementById("motionState");
483     const currentCO2 = document.getElementById("currentCO2");
484     const currentTime = document.getElementById("currentTime");
485     const energyPolicy = document.getElementById("energyPolicy");
486     const thermalZoneLevel = document.getElementById("thermalZoneLevel");
487     const proveBtn = document.getElementById("proveBtn");
488     const resultsHeader = document.getElementById("resultsHeader");
489     const resultsParagraph = document.getElementById("resultsParagraph");
490
491     function clearInputs (data) {
492         action.value = 'cool';
493         coolerState.checked = false;
494         heaterState.checked = false;
495         alarmState.checked = false;
496         currentTemp.value = null;
497         targetTemp.value = null;
498         motionState.checked = false;
499         currentCO2.value = null;
500         currentTime.value = null;
501         energyPolicy.value = 'none';
502         thermalZoneLevel.value = '1';
503         resultsHeader.hidden = true;
504         resultsParagraph.textContent = "";
505     }
506
507     async function fetchRoomState () {
508         let response = await fetch('/api/knx_baos_csv');
509         let data = await response.json();
510         console.log(data);
511         currentTemp.value = data['T'];
512         currentCO2.value = data['CO2'];
513         motionState.checked = data['illuminance'] >= 150;
514         const now = new Date();
515         currentTime.value = `${now.getHours()}:${now.getMinutes()}`;
516     }
517
518     function onSelectChange (el) {
519         console.log('changed!', el.value);
520         clearInputs();
521         if (el.value === 'room2') {
522             fetchRoomState();

```

```

523     }
524 }
525
526 async function prove () {
527     resultsHeader.hidden = true;
528     resultsParagraph.textContent = "";
529
530     const data = {
531         rooms: rooms.value,
532         action: action.value,
533         coolerState: coolerState.checked,
534         heaterState: heaterState.checked,
535         alarmState: alarmState.checked,
536         currentTemp: currentTemp.value,
537         targetTemp: targetTemp.value,
538         motionState: motionState.checked,
539         currentCO2: currentCO2.value,
540         currentTime: currentTime.value,
541         energyPolicy: energyPolicy.value,
542         thermalZoneLevel: thermalZoneLevel.value,
543     }
544
545     let response = await fetch('/api/gorgias', {
546         method: 'POST',
547         mode: 'cors', // no-cors, *cors, same-origin
548         cache: 'no-cache', // *default, no-cache, reload, force-cache, only-if-cached
549         credentials: 'same-origin', // include, *same-origin, omit
550         headers: {
551             'Content-Type': 'application/json'
552         },
553         redirect: 'follow', // manual, *follow, error
554         referrerPolicy: 'no-referrer', // no-referrer, *no-referrer-when-downgrade, origin, origin-when-
555 cross-origin, same-origin, strict-origin, strict-origin-when-cross-origin, unsafe-url
556         body: JSON.stringify(data) // body data type must match "Content-Type" header
557     });
558     let result = await response.json();
559     console.log(result);
560     resultsHeader.hidden = false;
561     resultsParagraph.textContent = result;
562 }
563 </script>
564 </html>

```

## 565 Gorgias files

### 566 Background policy

```

567 %space_121 auditorium
568 auditorium(space_121).
569 schedule(space_121,900,1800).
570 device(cooler1,cooler,space_121).

```

```

571 device(heater1,heater,space_121).
572 device(sensor1,sensor,space_121).
573 device(alarm1,alarm,space_121).
574
575 %space_110 user SENSOR ROOM
576 device(cooler2,cooler,space_110).
577 device(heater2,heater,space_110).
578 device(sensor2,sensor,space_110).
579 device(alarm2,alarm,space_110).
580
581 %space_115 computer room
582 computerRoom(space_115).
583 device(cooler3,cooler,space_115).
584 device(heater3,heater,space_115).
585 device(sensor3,sensor,space_115).
586 device(alarm3,alarm,space_115).
587
588 %space_112 user
589 device(cooler4,cooler,space_112).
590 device(heater4,heater,space_112).
591 device(sensor4,sensor,space_112).
592 device(alarm4,alarm,space_112).
593 %space_111 user
594 device(cooler5,cooler,space_111).
595 device(heater5,heater,space_111).
596 device(sensor5,sensor,space_111).
597 device(alarm5,alarm,space_111).
598
599 level(thermal_Zone_20,1).
600 level(thermal_Zone_21,2).
601 level(thermal_Zone_23,3).
602 level(thermal_Zone_18,4).
603 level(thermal_Zone_24,5).
604 level(thermal_Zone_29,6).
605 level(thermal_Zone_27,7).
606 level(thermal_Zone_17,8).
607 level(thermal_Zone_28,9).
608 level(thermal_Zone_25,10).
609 level(thermal_Zone_22,11).
610 level(thermal_Zone_19,12).
611
612 roomZone(space_121,thermal_Zone_23).
613 roomZone(space_110,thermal_Zone_20).
614 roomZone(space_115,thermal_Zone_18).
615 roomZone(space_112,thermal_Zone_20).
616 roomZone(space_111,thermal_Zone_20).
617 roomZone(space_114,thermal_Zone_18).
618 roomZone(space_113,thermal_Zone_18).
619 roomZone(space_120,thermal_Zone_23).
620 roomZone(space_122,thermal_Zone_21).
621 roomZone(space_123,thermal_Zone_21).
622 roomZone(space_125,thermal_Zone_22).
623 roomZone(space_133,thermal_Zone_17).

```

```

624 roomZone(space_134,thermal_Zone_25).
625 roomZone(space_137,thermal_Zone_24).
626 roomZone(space_147,thermal_Zone_27).
627 roomZone(space_145,thermal_Zone_28).
628 roomZone(space_148,thermal_Zone_27).
629 roomZone(space_162,thermal_Zone_24).
630 roomZone(space_151,thermal_Zone_30).
631 roomZone(space_159,thermal_Zone_17).
632 roomZone(space_160,thermal_Zone_17).
633 roomZone(space_161,thermal_Zone_24).
634 roomZone(space_163,thermal_Zone_24).
635 roomZone(space_164,thermal_Zone_24).
636 roomZone(space_165,thermal_Zone_27).
637 roomZone(space_166,thermal_Zone_27).
638 roomZone(space_169,thermal_Zone_29).
639 roomZone(space_171,thermal_Zone_29).
640 roomZone(space_170,thermal_Zone_29).
641 roomZone(space_109,thermal_Zone_19).
642
643 %general rules
644 device(Y,heater_cooler,X):-device(Y,cooler,X).
645 device(Y,heater_cooler,X):-device(Y,heater,X).
646 %energy management rules
647 noMotion(Room):- device(Dev,Type,Room),Type=sensor,reading(Dev,false).
648 savingMode(Room) :-
649 device(Dev,heater_cooler,Room),roomZone(Room,Zone),level(Zone,L),limit(AL,saving),L<AL.
650 tempOutOfSavingBounds(Room) :- hasTemp(Room,
651 Temp),limitByUser(Room,_,Type),Temp>26,Type=cool.
652 tempOutOfSavingBounds(Room) :- hasTemp(Room,
653 Temp),limitByUser(Room,_,Type),Temp<19,Type=heat.
654 criticalMode(Room):-
655 device(Dev,heater_cooler,Room),roomZone(Room,Zone),level(Zone,L),limit(AL,critical),L<AL.
656 %emergency rules
657 highCO2(Room):-co2Level(Room,X),X>2000.
658 %alert(Room):-smoke(Room).
659 alert(Room):-highCO2(Room).
660 %user rules
661 notWithinHours(Room):-
662 current_time(X),auditorium(Room),schedule(Room,START,FINISH),X<START.
663 notWithinHours(Room):-
664 current_time(X),auditorium(Room),schedule(Room,START,FINISH),X>FINISH.
665 tempOutOfBounds(Room) :- hasTemp(Room, Temp),
666 limitByUser(Room,X,Type),Temp>X,Type=cool.
667 tempOutOfBounds(Room) :- hasTemp(Room, Temp),
668 limitByUser(Room,X,Type),Temp<X,Type=heat.
669
670
670 User Policy
671 :-dynamic notWithinHours/1,tempOutOfBounds/1,hasTemp/2,limitByUser/3,withinHours/1, isOn/2,
672 isOff/2, device/3,auditorium/1,schedule/3,current_time/1.
673

```

```

674 rule(r11(Room, Device), switchOn(Room, Device), []):-
675 isOff(Device,Room),tempOutOfBounds(Room),device(Device,heater_cooler,Room).
676 rule(r12(Room, Device), neg(switchOn(Room, Device)), []):-
677 isOff(Device,Room),device(Device,heater_cooler,Room).
678 rule(p11(Room, Device), prefer(r11(Room, Device),r12(Room, Device)), []).
679 rule(p12(Room, Device), prefer(r12(Room, Device),r11(Room, Device)), []):-
680 notWithinHours(Room).
681 rule(c12(Room, Device), prefer(p12(Room, Device),p11(Room, Device)), []).
682
683 rule(r21(Room, Device), neg(switchOff(Room, Device)), []):-
684 isOn(Device,Room),tempOutOfBounds(Room),device(Device,heater_cooler,Room).
685 rule(r22(Room, Device), switchOff(Room, Device), []):-
686 isOn(Device,Room),device(Device,heater_cooler,Room).
687 rule(p21(Room, Device), prefer(r21(Room, Device),r22(Room, Device)), []).
688 rule(p22(Room, Device), prefer(r22(Room, Device),r21(Room, Device)), []):-
689 notWithinHours(Room).
690 rule(c22(Room, Device), prefer(p22(Room, Device),p21(Room, Device)), []).
691
692 complement(switchOn(Device, Room),switchOff(Device, Room)).
693 complement(switchOff(Device, Room),switchOn(Device, Room)).

694 Energy Management Policy

695 :-dynamic
696 noMotion/1,criticalMode/1,savingMode/1,device/3,reading/2,isOn/2,isOff/2,roomZone/2,level/2,limit/2,hasTemp/2,limitByUser/3,tempOutOfBounds/1.
697
698
699 %Policy for motion sensor
700 rule(r_mose(Room, Device), switchOff(Room, Device), []):-
701 noMotion(Room),isOn(Device,Room),device(Device,heater_cooler,Room).
702
703 %Policy for saving mode
704 rule(r11(Room, Device), switchOn(Room, Device), []):-
705 isOff(Device,Room),savingMode(Room),tempOutOfBounds(Room),device(Device,heater_cooler,Room).
706 rule(r12(Room, Device), neg(switchOn(Room, Device)), []):-
707 isOff(Device,Room),savingMode(Room),device(Device,heater_cooler,Room).
708 rule(p11(Room, Device), prefer(r11(Room, Device),r12(Room, Device)), []).
709 rule(p12(Room, Device), prefer(r12(Room, Device),r11(Room, Device)), []):-
710 notWithinHours(Room).
711 rule(c12(Room, Device), prefer(p12(Room, Device),p11(Room, Device)), []).
712
713 rule(r21(Room, Device), neg(switchOff(Room, Device)), []):-
714 isOn(Device,Room),savingMode(Room),tempOutOfBounds(Room),device(Device,heater_cooler,Room).
715 rule(r22(Room, Device), switchOff(Room, Device), []):-
716 isOn(Device,Room),savingMode(Room),device(Device,heater_cooler,Room).
717 rule(p21(Room, Device), prefer(r21(Room, Device),r22(Room, Device)), []).
718 rule(p22(Room, Device), prefer(r22(Room, Device),r21(Room, Device)), []):-
719 notWithinHours(Room).
720 rule(c22(Room, Device), prefer(p22(Room, Device),p21(Room, Device)), []).
721
722 %Policy for critical mode
723
724

```

```

725 rule(r_crmo(Room, Device), switchOff(Room, Device), []):-
726 criticalMode(Room),isOn(Device,Room),device(Device,heater_cooler,Room).
727
728 %combination motion sensor, saving
729 rule(p_mose_saving(Room, Device), prefer(r_mose(Room, Device),r21(Room, Device)), []).
730
731 complement(switchOff(Room, Device), switchOn(Room, Device)).
732 complement(switchOn(Room, Device), switchOff(Room, Device)).
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771

```

Emergency Policy

```

:-dynamic isOn/2, isOff/2, smoke/1, alert/1, highCO2/1.

rule(r_em1(Device, Room), switchOff(Room, Device),[]):-isOn(Device,Room),alert(Room).
rule(r_em4(Device, Room), neg(switchOff(Room, Device)),[]):-
isOn(Device,Room),alert(Room),device(Device,alarm,Room).
rule(r_em2(Device, Room), neg(switchOn(Room, Device)),[]):-isOff(Device,Room),alert(Room).
rule(r_em3(Device, Room), switchOn(Room,Device),[]):-
isOff(Device,Room),alert(Room),device(Device,alarm,Room).

rule(p_em32(Device, Room), prefer(r_em3(Device, Room),r_em2(Device, Room)), []).
rule(p_em41(Device, Room), prefer(r_em4(Device, Room),r_em1(Device, Room)), []).

complement(switchOn(Device, Room),switchOff(Device, Room)).
complement(switchOff(Device, Room),switchOn(Device, Room)).

```

Meta-Data Policy

```

:- dynamic computerRoom/1.

rule(r1(personal), policy(personal), []).
rule(r2(management), policy(management), []).
rule(r3(emergency), policy(emergency), []).

rule(p1(emergency), prefer(r3(emergency), r1(personal)), []).
rule(p2(emergency), prefer(r3(emergency), r2(management)), []).
rule(p3(management), prefer(r2(management), r1(personal)), []).
rule(p4(personal), prefer(r1(personal), r2(management)), []):-
device(Device,cooler,Room),computerRoom(Room).
rule(c1(personal), prefer(p4(personal), p3(management)), []).

complement(policy(emergency), policy(management)).
complement(policy(management), policy(emergency)).
complement(policy(emergency), policy(personal)).
complement(policy(personal), policy(emergency)).
complement(policy(management), policy(personal)).
complement(policy(personal), policy(management)).

```

772

## Combined Policy

```

773 :-dynamic computerRoom/1, isOn/2, isOff/2, smoke/1, alert/1, highCO2/1, noMotion/1,
774 criticalMode/1, savingMode/1, device/3, reading/2, roomZone/2, level/2, limit/2, hasTemp/2,
775 limitByUser/3, tempOutOfSavingBounds/1, notWithinHours/1, tempOutOfBounds/1, auditorium/1,
776 schedule/3, current_time/1, co2Level/2.
777
778 %user local policy
779
780 rule(user_r11(Room, Device, user), switchOn_local(Room, Device, user), []):-isOff(Device,
781 Room), tempOutOfBounds(Room), device(Device, heater_cooler, Room).
782 rule(user_r12(Room, Device, user), neg(switchOn_local(Room, Device, user)), []):-isOff(Device,
783 Room), device(Device, heater_cooler, Room).
784 rule(user_p11(Room, Device, user), prefer(user_r11(Room, Device, user), user_r12(Room,
785 Device, user)), []).
786 rule(user_p12(Room, Device, user), prefer(user_r12(Room, Device, user), user_r11(Room,
787 Device, user)), []):-notWithinHours(Room).
788 rule(user_c12(Room, Device, user), prefer(user_p12(Room, Device, user), user_p11(Room,
789 Device, user)), []).
790
791 rule(user_r21(Room, Device, user), neg(switchOff_local(Room, Device, user)), []):-isOn(Device,
792 Room), tempOutOfBounds(Room), device(Device, heater_cooler, Room).
793 rule(user_r22(Room, Device, user), switchOff_local(Room, Device, user), []):-isOn(Device,
794 Room), device(Device, heater_cooler, Room).
795 rule(user_p21(Room, Device, user), prefer(user_r21(Room, Device, user), user_r22(Room,
796 Device, user)), []).
797 rule(user_p22(Room, Device, user), prefer(user_r22(Room, Device, user), user_r21(Room,
798 Device, user)), []):-notWithinHours(Room).
799 rule(user_c22(Room, Device, user), prefer(user_p22(Room, Device, user), user_p21(Room,
800 Device, user)), []).
801
802 %energy local policy
803
804 rule(energy_r_mose1(Room, Device, energy), switchOff_local(Room, Device, energy), []):-
805 noMotion(Room), isOn(Device, Room), device(Device, heater_cooler, Room).
806 rule(energy_r_mose2(Room, Device, energy), neg(switchOn_local(Room, Device, energy)), []):-
807 noMotion(Room), isOff(Device, Room), device(Device, heater_cooler, Room).
808
809 rule(energy_r11(Room, Device, energy), switchOn_local(Room, Device, energy), []):-
810 isOff(Device, Room), savingMode(Room), tempOutOfSavingBounds(Room), device(Device,
811 heater_cooler, Room).
812 rule(energy_r12(Room, Device, energy), neg(switchOn_local(Room, Device, energy)), []):-
813 isOff(Device, Room), savingMode(Room), device(Device, heater_cooler, Room).
814 rule(energy_p11(Room, Device, energy), prefer(energy_r11(Room, Device, energy),
815 energy_r12(Room, Device, energy)), []).
816 rule(energy_p12(Room, Device, energy), prefer(energy_r12(Room, Device, energy),
817 energy_r11(Room, Device, energy)), []):-notWithinHours(Room).
818 rule(energy_c12(Room, Device, energy), prefer(energy_p12(Room, Device, energy),
819 energy_p11(Room, Device, energy)), []).
820
821 rule(energy_r21(Room, Device, energy), neg(switchOff_local(Room, Device, energy)), []):-
822 isOn(Device, Room), savingMode(Room), tempOutOfSavingBounds(Room), device(Device,
823 heater_cooler, Room).

```

```

824 rule(energy_r22(Room, Device, energy), switchOff_local(Room, Device, energy), []):-
825 isOn(Device, Room), savingMode(Room), device(Device, heater_cooler, Room).
826 rule(energy_p21(Room, Device, energy), prefer(energy_r21(Room, Device, energy),
827 energy_r22(Room, Device, energy)), []).
828 rule(energy_p22(Room, Device, energy), prefer(energy_r22(Room, Device, energy),
829 energy_r21(Room, Device, energy)), []):-notWithinHours(Room).
830 rule(energy_c22(Room, Device, energy), prefer(energy_p22(Room, Device, energy),
831 energy_p21(Room, Device, energy)), []).
832
833 rule(energy_r_crmo1(Room, Device, energy), switchOff_local(Room, Device, energy), []):-
834 criticalMode(Room), isOn(Device, Room), device(Device, heater_cooler, Room).
835 rule(energy_r_crmo2(Room, Device, energy), neg(switchOn_local(Room, Device, energy)), []):-
836 criticalMode(Room), isOff(Device, Room), device(Device, heater_cooler, Room).
837
838 rule(energy_p_mose_saving1(Room, Device, energy), prefer(energy_r_mose1(Room, Device,
839 energy),energy_r21(Room, Device, energy)), []).
840 rule(energy_p_mose_saving2(Room, Device, energy), prefer(energy_r_mose2(Room, Device,
841 energy),energy_r11(Room, Device, energy)), []).
842
843 %emergency local policy
844
845 rule(emergency_r_em1(Room, Device, emergency), switchOff_local(Room, Device, emergency),
846 []):-isOn(Device, Room), alert(Room).
847 rule(emergency_r_em4(Room, Device, emergency), neg(switchOff_local(Room, Device,
848 emergency)), []):-isOn(Device, Room), alert(Room),device(Device,alarm,Room).
849 rule(emergency_r_em2(Room, Device, emergency), neg(switchOn_local(Room, Device,
850 emergency)), []):-isOff(Device, Room), alert(Room).
851 rule(emergency_r_em3(Room, Device, emergency), switchOn_local(Room, Device, emergency),
852 []):-isOff(Device, Room), alert(Room),device(Device,alarm,Room).
853
854 rule(emergency_p_em32(Room, Device, emergency), prefer(emergency_r_em3(Room, Device,
855 emergency),emergency_r_em2(Room, Device, emergency)), []).
856 rule(emergency_p_em41(Room, Device, emergency), prefer(emergency_r_em4(Room, Device,
857 emergency),emergency_r_em1(Room, Device, emergency)), []).
858
859 %metapolicy
860
861 rule(r1(Room, Device, Policy1), switchOn(Room, Device), [switchOn_local(Room, Device,
862 Policy1)]).
863 rule(r2(Room, Device, Policy2), neg(switchOn(Room, Device)), [neg(switchOn_local(Room,
864 Device, Policy2))]).
865 rule(r3(Room, Device, Policy3), neg(switchOff(Room, Device)), [neg(switchOff_local(Room,
866 Device, Policy3))]).
867 rule(r4(Room, Device, Policy4), switchOff(Room, Device), [switchOff_local(Room, Device,
868 Policy4)]).
869
870 rule(p_em_pe12(emergency, Room, Device, user), prefer(r1(Room, Device, emergency),
871 r2(Room, Device, user)), []).
872 rule(p_em_pe21(emergency, Room, Device, user), prefer(r2(Room, Device, emergency),
873 r1(Room, Device, user)), []).
874 rule(p_em_pe34(emergency, Room, Device, user), prefer(r3(Room, Device, emergency),
875 r4(Room, Device, user)), []).
876 rule(p_em_pe43(emergency, Room, Device, user), prefer(r4(Room, Device, emergency),

```



```

877 r3(Room, Device, user)), []).
878
879 rule(p_em_en12(emergency, Room, Device, energy), prefer(r1(Room, Device, emergency),
880 r2(Room, Device, energy)), []).
881 rule(p_em_en21(emergency, Room, Device, energy), prefer(r2(Room, Device, emergency),
882 r1(Room, Device, energy)), []).
883 rule(p_em_en34(emergency, Room, Device, energy), prefer(r3(Room, Device, emergency),
884 r4(Room, Device, energy)), []).
885 rule(p_em_en43(emergency, Room, Device, energy), prefer(r4(Room, Device, emergency),
886 r3(Room, Device, energy)), []).
887
888 rule(p_en_pe12(energy, Room, Device, user), prefer(r1(Room, Device, energy), r2(Room,
889 Device, user)), []).
890 rule(p_en_pe21(energy, Room, Device, user), prefer(r2(Room, Device, energy), r1(Room,
891 Device, user)), []).
892 rule(p_en_pe34(energy, Room, Device, user), prefer(r3(Room, Device, energy), r4(Room,
893 Device, user)), []).
894 rule(p_en_pe43(energy, Room, Device, user), prefer(r4(Room, Device, energy), r3(Room,
895 Device, user)), []).
896
897 rule(p_pe_en12(user, Room, Device, energy), prefer(r1(Room, Device, user), r2(Room, Device,
898 energy)), []):-computerRoom(Room).
899 rule(p_pe_en21(user, Room, Device, energy), prefer(r2(Room, Device, user), r1(Room, Device,
900 energy)), []):-computerRoom(Room).
901 rule(p_pe_en34(user, Room, Device, energy), prefer(r3(Room, Device, user), r4(Room, Device,
902 energy)), []):-computerRoom(Room).
903 rule(p_pe_en43(user, Room, Device, energy), prefer(r4(Room, Device, user), r3(Room, Device,
904 energy)), []):-computerRoom(Room).
905
906 rule(c111(user, Room, Device, energy), prefer(p_pe_en12(user, Room, Device, energy),
907 p_en_pe21(energy, Room, Device, user)), []).
908 rule(c121(user, Room, Device, energy), prefer(p_pe_en21(user, Room, Device, energy),
909 p_en_pe12(energy, Room, Device, user)), []).
910 rule(c111(user, Room, Device, energy), prefer(p_pe_en34(user, Room, Device, energy),
911 p_en_pe43(energy, Room, Device, user)), []).
912 rule(c121(user, Room, Device, energy), prefer(p_pe_en43(user, Room, Device, energy),
913 p_en_pe34(energy, Room, Device, user)), []).
914
915 %complement(switchOn_local(Room, Device, Policy1), switchOff_local(Room, Device, Policy2)).
916 %complement(switchOff_local(Room, Device, Policy1), switchOn_local(Room, Device, Policy2)).
917
918 %complement(switchOff(X, Y, Z1), neg(switchOff(X, Y, Z2))).
919 %complement(switchOn(X, Y, Z1), neg(switchOff(X, Y, Z2))).
920
921
922
923
924

```