

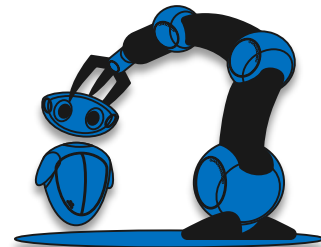


ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Εργαστήριο Επεξεργασίας Πληροφορία και Υπολογισμών



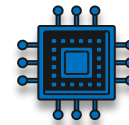
# Σχεδίαση και ανάπτυξη Μηχανισμού Αυτοματοποίησης της παραμετροποίησης ρομποτικών συσκευών για το περιβάλλον προσομοίωσης Gazebo

---

*Design and implementation of an Automation Mechanism for the configuration  
of robotic devices for the Gazebo simulator*

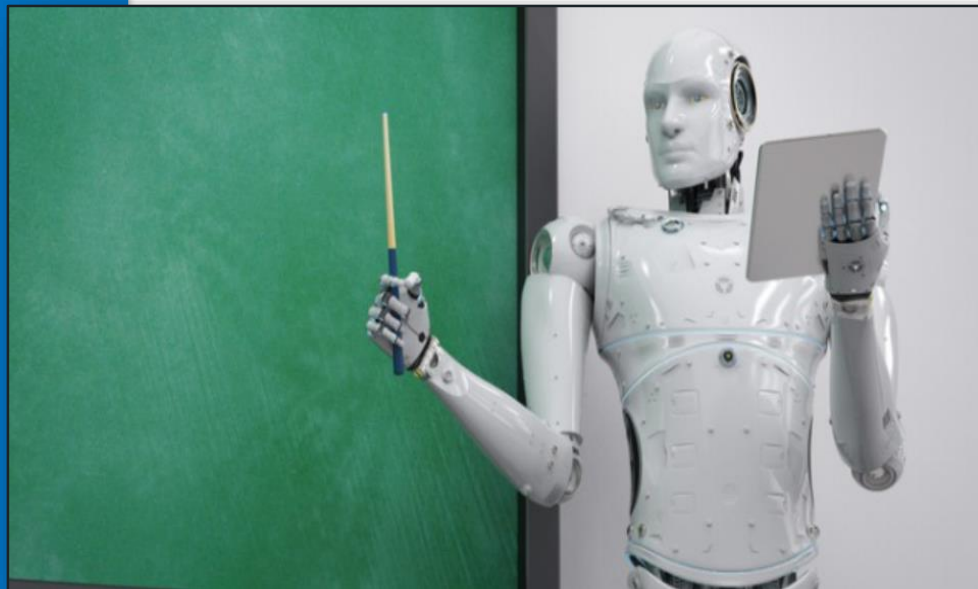
Διπλωματική Εργασία του  
Χωραφά Χρήστου  
Α.Ε.Μ: 8718

# Λογισμικό και Ρομπότ



Η άνθιση της τεχνολογίας έχει “εγκαταστήσει” στην ζωή μας το λογισμικό και την ρομποτική:

- Υπάρχει μεγάλη **ανάγκη** για κατασκευή ρομπότ και λογισμικού για ρομπότ.
- Στις μέρες μας, οι περισσότερες δοκιμές ρομπότ και εφαρμογών για αυτά γίνονται σε ειδικά λογισμικά, τους **Ρομποτικούς Προσομοιωτές**.
- **Μειώνεται** έτσι σημαντικά το κόστος και ο χρόνος ανάπτυξης.



# Προβλήματα



Παρά τα οφέλη τους, οι προσομοιώσεις φέρουν **δυσκολίες**:

1

Στο *Gazebo*, η **παραμετροποίηση** ενός στοιχείου του απαιτεί την **σύνταξη** ενός **.sdf** αρχείου, που περιγράφει αυτό το στοιχείο.

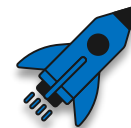
2

Πράγμα **δύσκολο**, αφού η σύνταξη ενός τέτοιου αρχείου απαιτεί **ενδελεχή μελέτη** των **προδιαγραφών** του *SDF* και γνώση της *XML*.

3

Πολλοί μηχανισμοί παράγουν κώδικα για ρομπότ, αλλά **δεν διαθέτουν** προσομοιωτές ή δυνατότητα προσομοίωσης.

# Λύση – Αυτοματοποίηση Λογισμικού



✓ Ταυτόχρονα, τις τελευταίες δύο δεκαετίες αναπτύσσεται ραγδαία ο κλάδος της **Αυτοματοποιημένης Μηχανικής Λογισμικού**.

✓ Αυξάνεται η **παραγωγικότητα** και **ποιότητα** λογισμικού.



✓ Προσφέρει **υψηλότερο** αφαιρετικό επίπεδο, ώστε να επικεντρώνεται ο μηχανικός στο πρόβλημα και **όχι** στις λεπτομέρειες υλοποίησης.

✓ Συνδυάζεται με μια μεθοδολογία αυτοματοποίησης της μηχανικής λογισμικού, την **Μοντελοστραφή Μηχανική**.



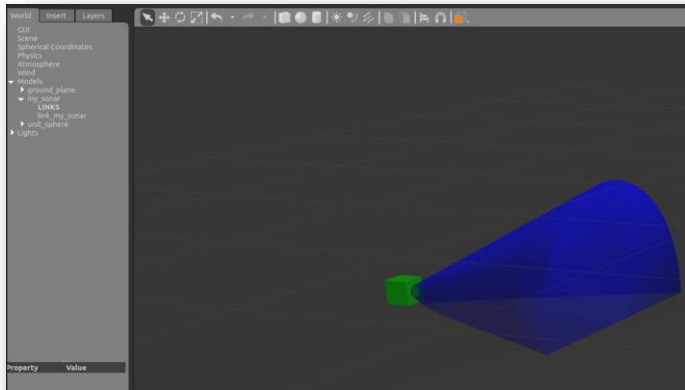
# Μηχανισμός Αυτοματοποίησης

## Πιο εύκολη/γρήγορη προσομοίωση

Δεν απαιτεί τεχνικές γνώσεις



## Παραγωγή αρχείων αισθητήρων για τον προσομοιωτή *Gazebo*

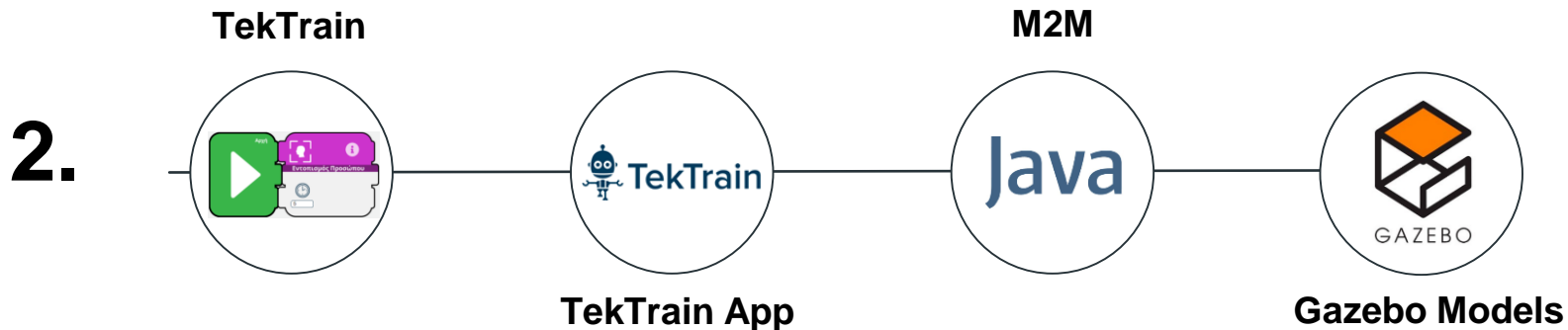
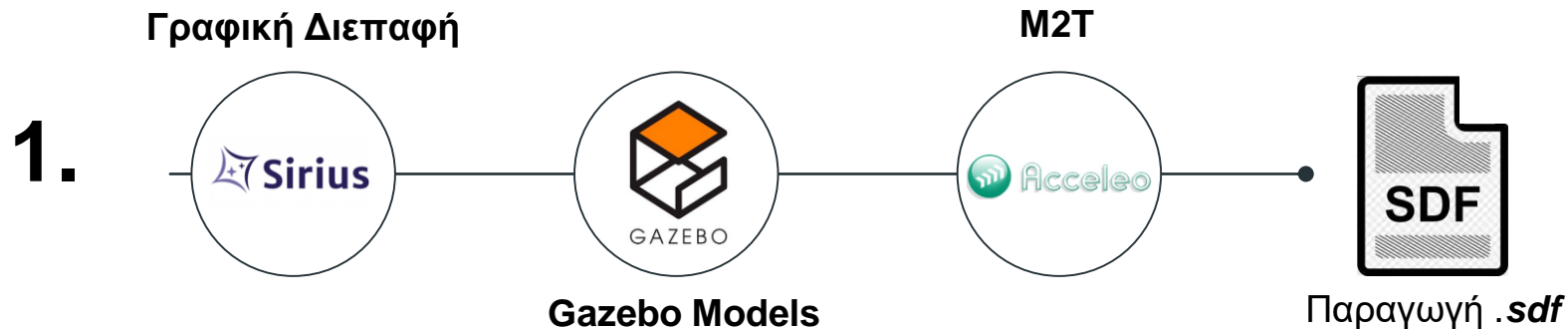


## Αυτοματοποίηση προσομοίωσης για εφαρμογές ρομποτικής TekTrain



# Στόχοι Διπλωματικής Εργασίας

# Ροές Διπλωματικής





# Βήματα Ανάπτυξης

- Μελέτη του ***SDFormat***.

Συλλογή προδιαγραφών που ορίζουν την σύνταξη ενός .sdf αρχείου.

- Δημιουργία **μετα-μοντέλου**.

Προκύπτει από το SDFormat, περιέχει την πληροφορία των μοντέλων.

- Υλοποίηση **γραφικής διεπαφής**

Μέσω του Sirius UI.

- Δημιουργία **Μετασχηματισμού M2T**

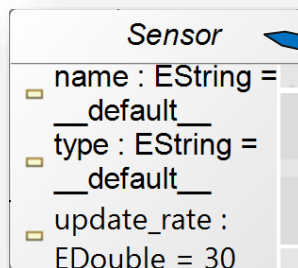
Μετασχηματισμός μοντέλων σε αρχεία .sdf.

- Δημιουργία **Μετασχηματισμού M2M**

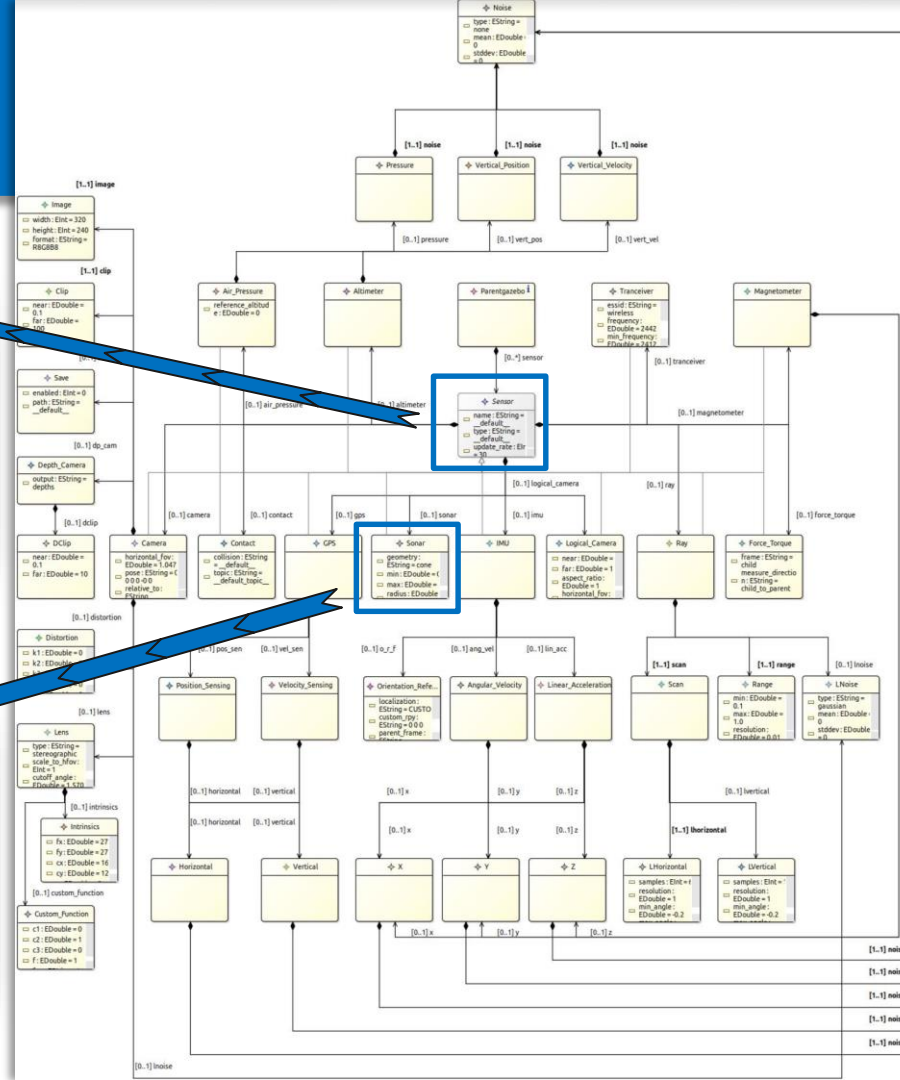
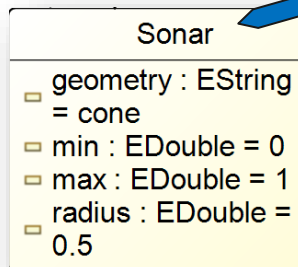
Μετασχηματισμός εφαρμογής TekTrain σε Gazebo Model.

[illegible]

1. Η κλάση *Sensor*. Περιέχει τα βασικά χαρακτηριστικά, που κληρονομεί ο κάθε αισθητήρας.

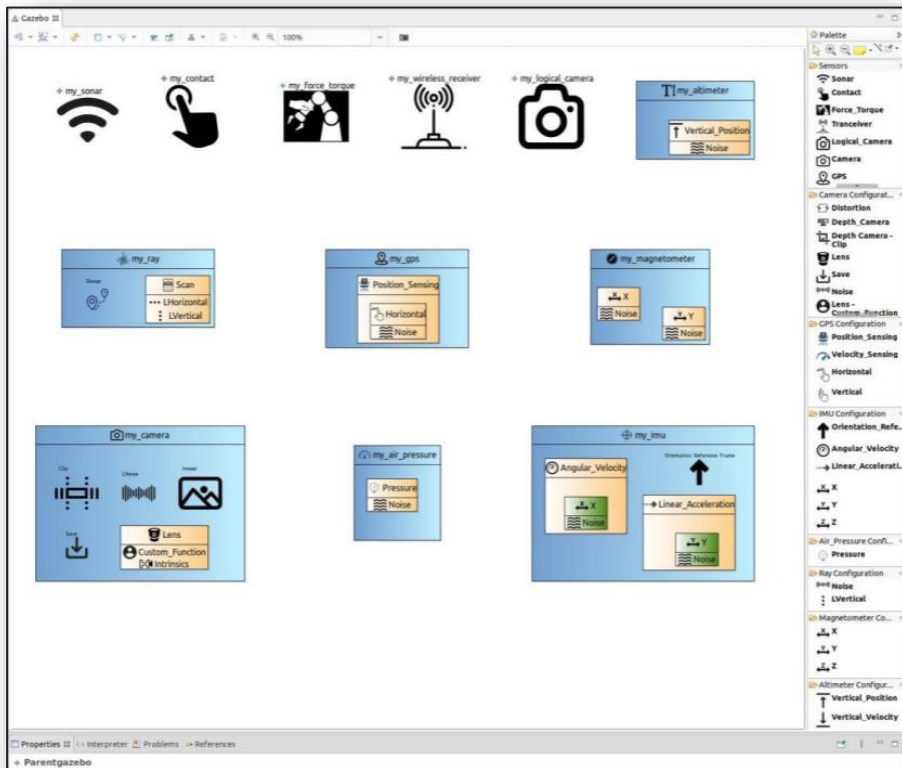


2. Για παράδειγμα ο αισθητήρας Sonar, πέρα από τα δικά του ξεχωριστά χαρακτηριστικά, κληρονομεί και τα βασικά της κλάσης *Sensor*.





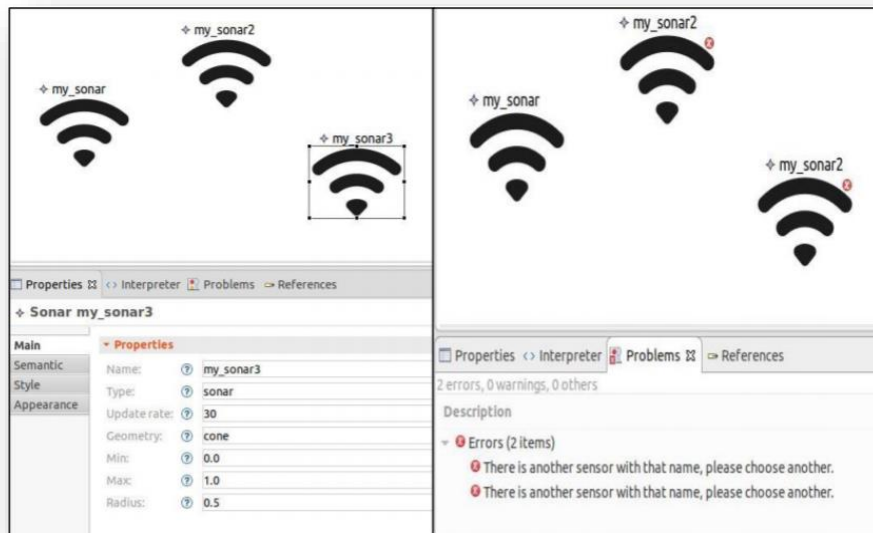
# Γραφική Διεπαφή



Στην γραφική διεπαφή οι αισθητήρες χωρίζονται σε τύπους **Node** και **Container**. *Nodes* είναι οι αισθητήρες που δεν δέχονται έξτρα λειτουργίες, σε αντίθεση με τα *Containers*. Οι αισθητήρες *Nodes* απεικονίζονται με το εικονίδιο τους, ενώ οι αισθητήρες *Containers* απεικονίζονται ως “boxes”, μέσα στα οποία μπορεί ο χρήστης να εισάγει περαιτέρω λειτουργικότητα.

Στην δεξιά πλευρά βρίσκεται η **παλέτα** της διεπαφής και περιλαμβάνει όλους τους αισθητήρες και τις έξτρα λειτουργικότητες των αισθητήρων τύπου *Container*.

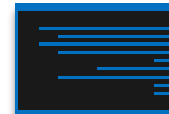
# Γραφική Διεπαφή



Στην γραφική διεπαφή ορίζονται και κάποιοι **Κανόνες Επικύρωσης**, που διασφαλίζουν την σωστή μοντελοποίηση αισθητήρων, χωρίς σφάλματα.

Ενδεικτικά, αριστερά παρουσιάζεται ένας κανόνας που εξασφαλίζει την μοναδική ονομασία των αισθητήρων.

# Παραδείγματα μοντέλων αισθητήρων



*Τα Gazebo Models Camera και Sonar, που δημιουργήθηκαν στο GUI.*

*Δεξιά απεικονίζονται οι παραμετροποιήσεις, που μπορούν να δεχτούν.*

Όνομα – Τύπος	Εικονίδιο	Απεικόνιση Χαρακτηριστικών
Camera Sensor – Container		<div>▼ Properties</div> <div><div>Name:</div><div><input type="text" value="my_camera"/></div><div>Type:</div><div><input type="text" value="camera"/></div><div>Update rate:</div><div><input type="text" value="1"/></div><div>Horizontal fov:</div><div><input type="text" value="1.047"/></div><div>Pose:</div><div><input type="text" value="0.1 0 0 0 0 0"/></div><div>Relative to:</div><div><input type="text" value="parent"/></div></div>

Όνομα – Τύπος	Εικονίδιο	Απεικόνιση Χαρακτηριστικών
Sonar Sensor – Node		<div>▼ Properties</div> <div><div>Name:</div><div><input type="text" value="my_sonar"/></div><div>Type:</div><div><input type="text" value="sonar"/></div><div>Update rate:</div><div><input type="text" value="30"/></div><div>Geometry:</div><div><input type="text" value="cone"/></div><div>Min:</div><div><input type="text" value="0.0"/></div><div>Max:</div><div><input type="text" value="1.0"/></div><div>Radius:</div><div><input type="text" value="0.5"/></div></div>

# Model To Text – M2T



```
<sensor name='{s.name/}' type='{s.type/}'>
  <[s.type.toLower()/]>
  [for (att : EAttribute | s.eClass().eAttributes)]
    [if (att.name = 'pose')]
      <pose relative_to='{s.eGet('relative to')/}'>[s.eGet('relative to')]</pose>
    [elseif (att.name = 'relative to')]<[else]>
      <[att.name/]>[s.eGet(name)/]</[att.name/]>
    [/if]
  [/for]
  [if (s.eContents()->isEmpty() = false)] [comment "Means
  [for (subCll : OclAny | s.eContents())]
    [let className : EString = subCll.eClass().name
    [if (subCll.isKindOf(EClass))] [comment "diffuse"]
```



```
<?xml version='1.0'?>
<sdf version='1.7'>
  <model name='my_sonar'>
    <pose>0 0 0 0 -1.5707 0</pose>
    <link name='link_my_sonar'>
      <visual name='visual'>
        <geometry>
          <box>
            <size>0.1 0.1 0.1</size>
          </box>
        </geometry>
        <material>
          <ambient>0.1 1 0.1 1</ambient>
          <diffuse>0.3 0.9 0.1 1</diffuse>
          <specular>0 0 0 0</specular>
          <emissive>0 0 0 1</emissive>
        </material>
      </visual>
      <collision name='collision'>
        <geometry>
          <box>
            <size>0.1 0.1 0.1</size>
          </box>
        </geometry>
      </collision>
      <sensor name='my_sonar' type='sonar'>
        <sonar>
          <geometry>cone</geometry>
          <min>0.0</min>
          <max>1.0</max>
          <radius>0.5</radius>
        </sonar>
        <always_on>1</always_on>
        <update_rate>30</update_rate>
        <visualize>true</visualize>
      </sensor>
    </link>
  </model>
</sdf>
```

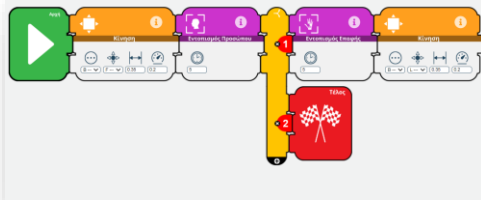


Εισαγωγή του δημιουργούμενου μοντέλου Sonar από την γραφική διεπαφή στον **M2T** μετασχηματισμό και παραγωγή του **.sdf** αρχείου του.

# Model To Model – M2M

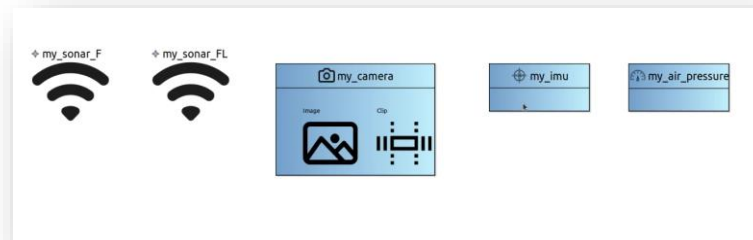


## TekTrain App



```
if (temp instanceof CameraMotion || temp instanceof DetectFace ||  
    || temp instanceof DetectBarcode || temp instanceof Detect  
    || temp instanceof DetectMotion || temp instanceof Detect  
    || temp instanceof DetectAngle) {  
    if (!checkCan) {  
        Camera camera = GazeboFactory.eINSTANCE.createCamera();  
        camera.setImage(GazeboFactory.eINSTANCE.createImage());  
        camera.setClip(GazeboFactory.eINSTANCE.createClip());  
        camera.setName("my_camera");  
        camera.setType("camera");  
        gazebo.getSensor().add(camera);  
        checkCan = true;  
    }  
}
```

Java



Εισαγωγή μιας εφαρμογής TekTrain στον **M2M**  
μετασχηματισμό και παραγωγή των αντίστοιχων Gazebo  
Models.

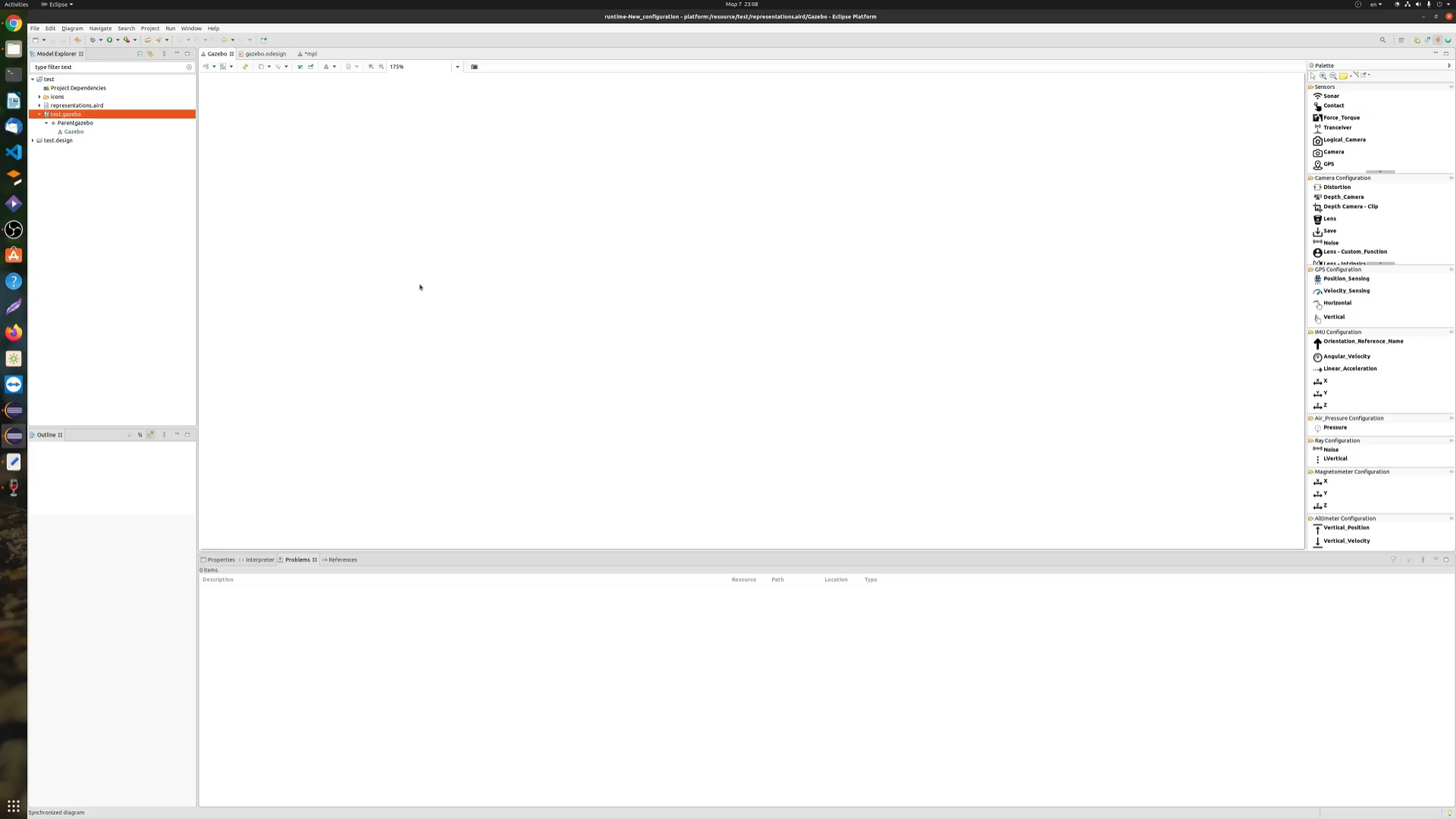
**Παραδοχή:** δημιουργείται **μόνο** ένα μοντέλο αισθητήρα για  
κάθε σύνολο λειτουργικότητας, **εκτός** από το Sonar, που  
δημιουργείται ένα για κάθε κατεύθυνση που ανίχνευσης  
εμποδίου.

# Παράδειγμα Χρήσης 1



*Ακολουθεί ένα βίντεο, στο οποίο παρουσιάζεται η Ροή 1.*

*Δημιουργία μοντέλων αισθητήρων μέσω της γραφικής  
διεπαφής και μετασχηματισμός τους σε .sdf αρχεία.*



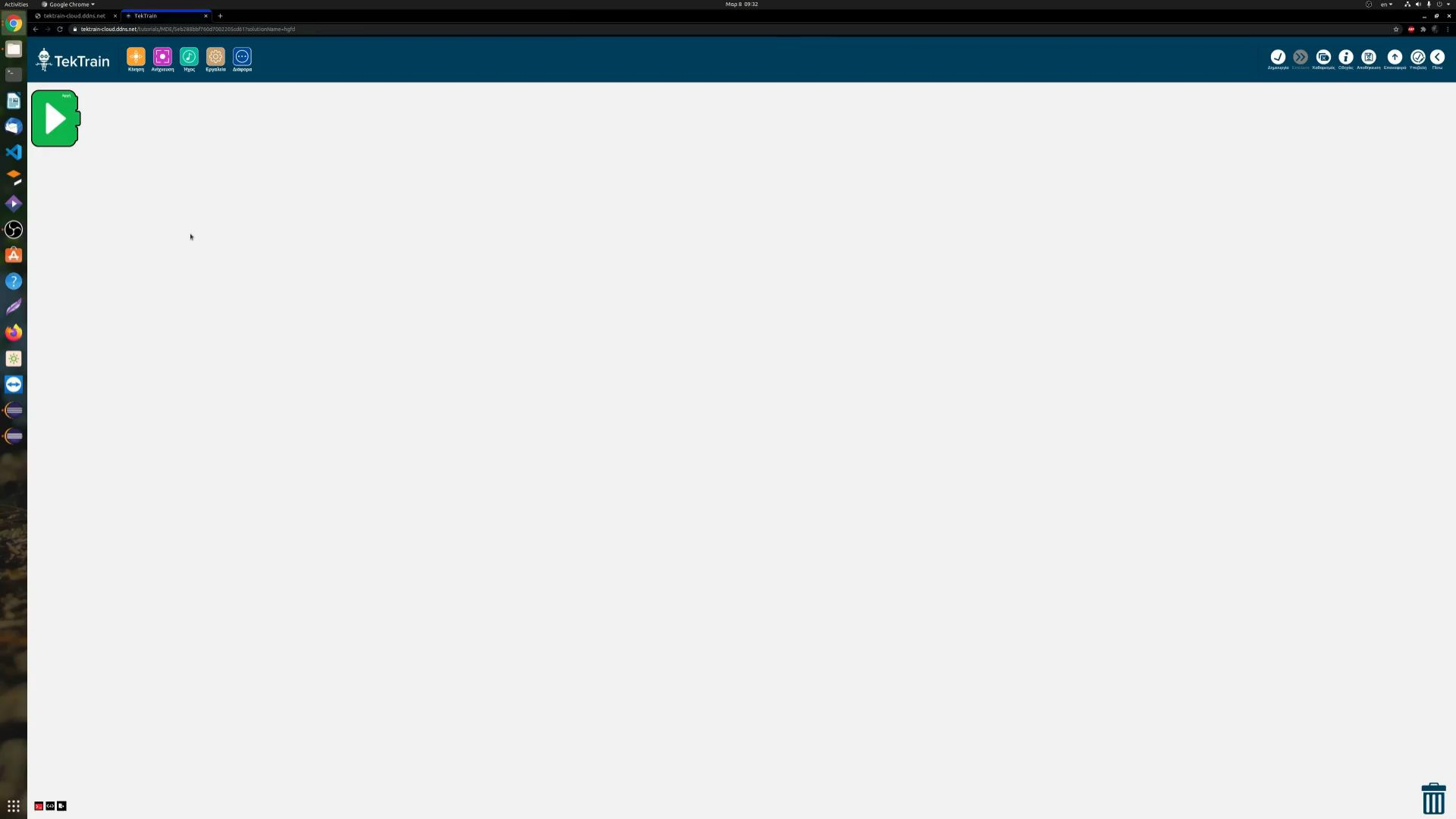
# Παράδειγμα Χρήσης 2



*Ακολουθεί ένα βίντεο, στο οποίο παρουσιάζεται η Ροή 2.*

*Δημιουργία μιας εφαρμογής TekTrain και ο μετασχηματισμός της σε Gazebo Models.*



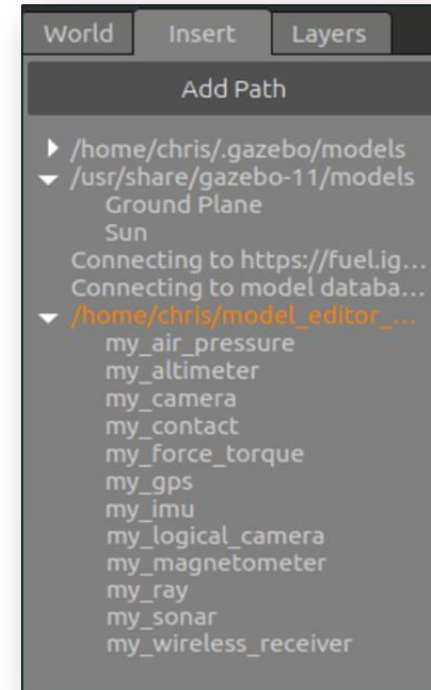


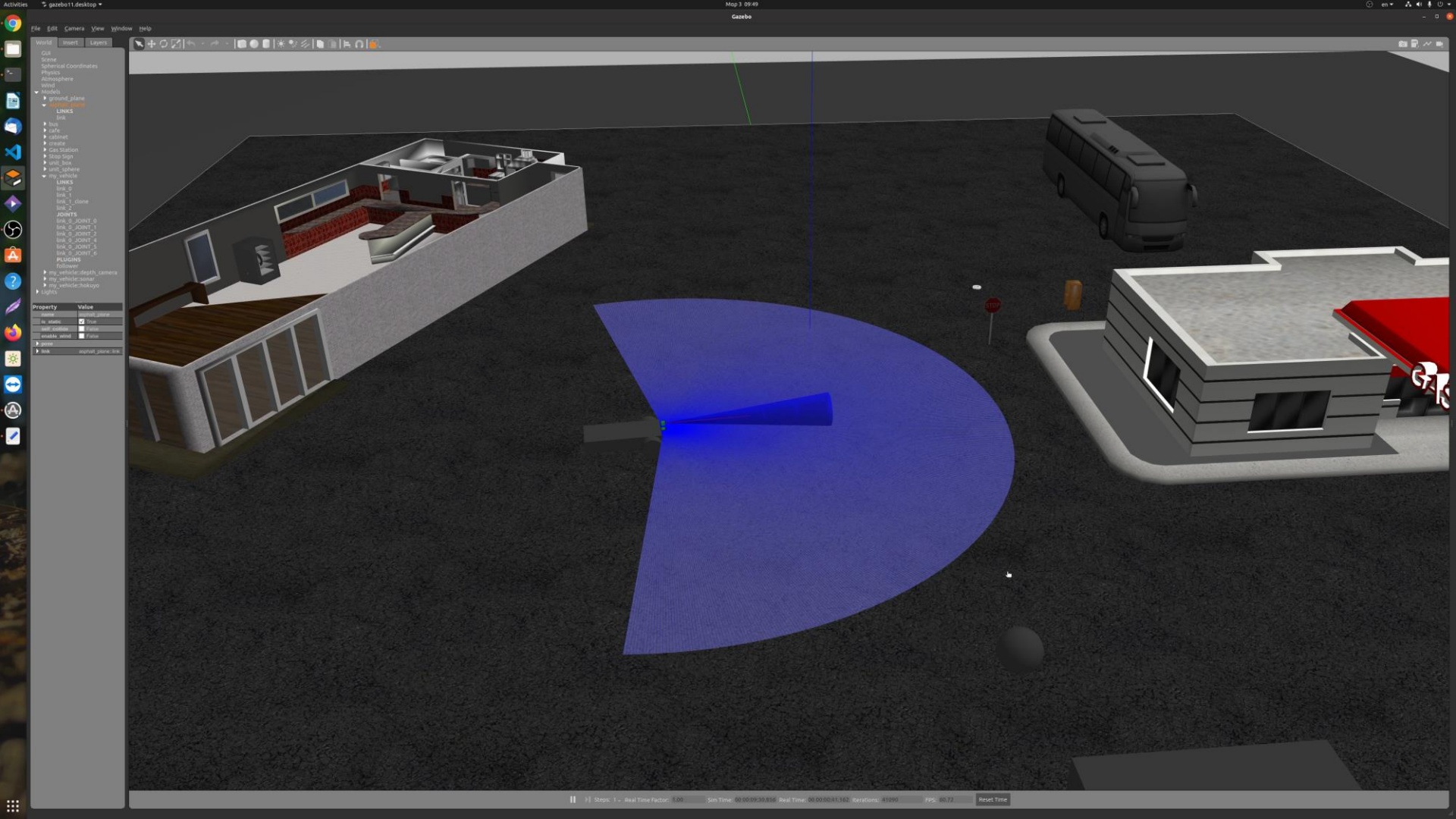
# Ενδεικτική Προσομοίωση



Αφού γίνει η **εισαγωγή** των παραγόμενων *.sdf* αρχείων στο *Gazebo*, μπορούμε να προχωρήσουμε στην προσομοίωση.

Ακολουθεί ένα βίντεο, στο οποίο παρουσιάζεται μία προσομοίωση στο *Gazebo* με ένα όχημα. Φέρει πάνω του τους αισθητήρες *Sonar* και *Ray*, που δημιουργήθηκαν με τον μηχανισμό αυτοματοποίησης.







## Συμπεράσματα

### Οι χρήστες **δεν** χρειάζεται

- Να γνωρίζουν την XML.
- Ούτε (σημαντικότερα) τη σύνταξη ενός **.sdf** αρχείου, που περιέχει τους αισθητήρες και τις παραμέτρους τους.

### Οι χρήστες **μπορούν**

- Να φτιάξουν κάποιο μοντέλο σε μια άλλη γραφική εφαρμογή, στην οποία είναι ήδη εξοικειωμένοι, και απλά να μετασχηματίσουν στην συνέχεια το μοντέλο τους σε μοντέλο συμβατό με το Gazebo.
- Να προχωρήσουν απευθείας στην προσομοίωση, μειώνοντας την πολυπλοκότητα και τα σφάλματα.

Σας ευχαριστώ για τον  
χρόνο σας!

---