Modern Robotic Systems

No robot is an island

- The robot is only a small part of a **robotic system.**

Kiva */ Amazon Robotics

* Raffaello D'Andrea

Robotic systems

- The **robot**:
  - The hardware: …
  - The software: …
- The **other robots**
- The **other machines**
- The **infrastructure**:
  - The network
  - The power
  - …

- The **people**, including:
  - Supervisors
  - Safety operators
  - Customers
  - Data annotators
  - QA
  - …

Classical robotics development

- The old model of development:
  - design
  - product development
  - integration ("system integrators")
  - installation
  - support

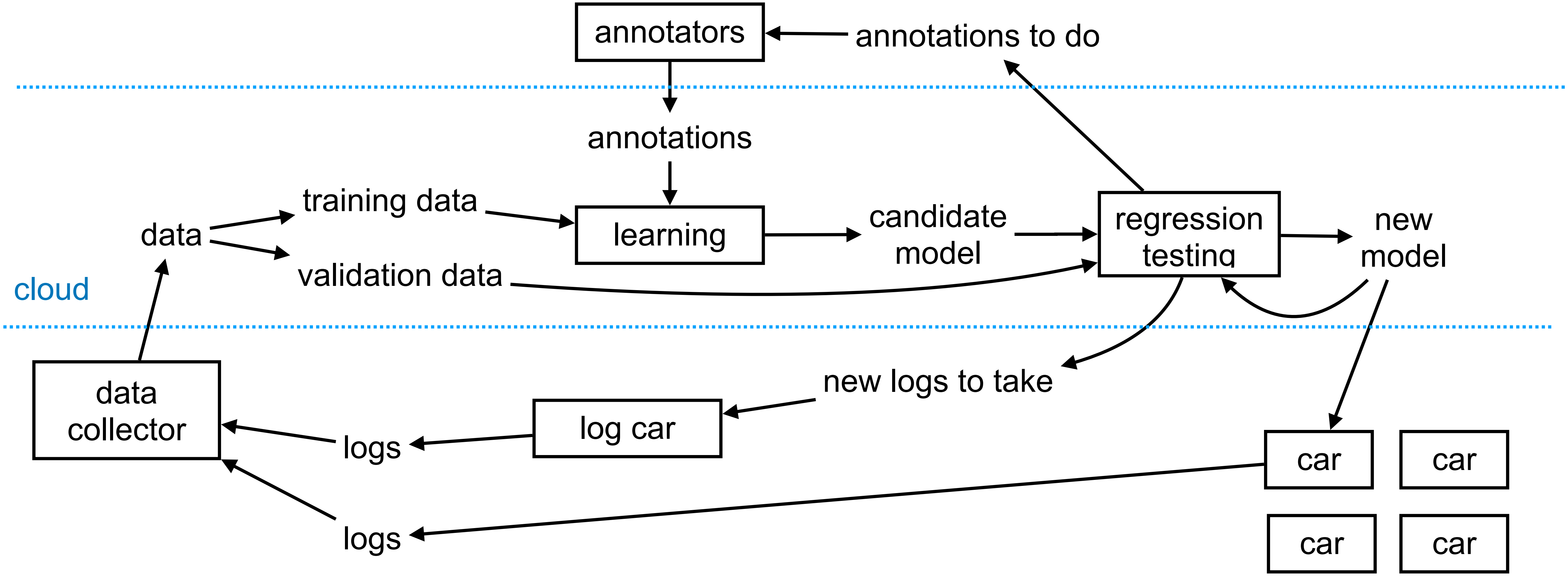
Modern robotics development

- Aspects of modern robotics development:
    - The system is **continuously evolving**.
    - **Agile** development.
    - **Continuous integration**.
    - There is continuous **feedback from deployment** (sensors, user feedback)
    - Great importance of **data**.

- We will see some of these "**best practices**" in the class.

Example of cloud pipeline for object detection

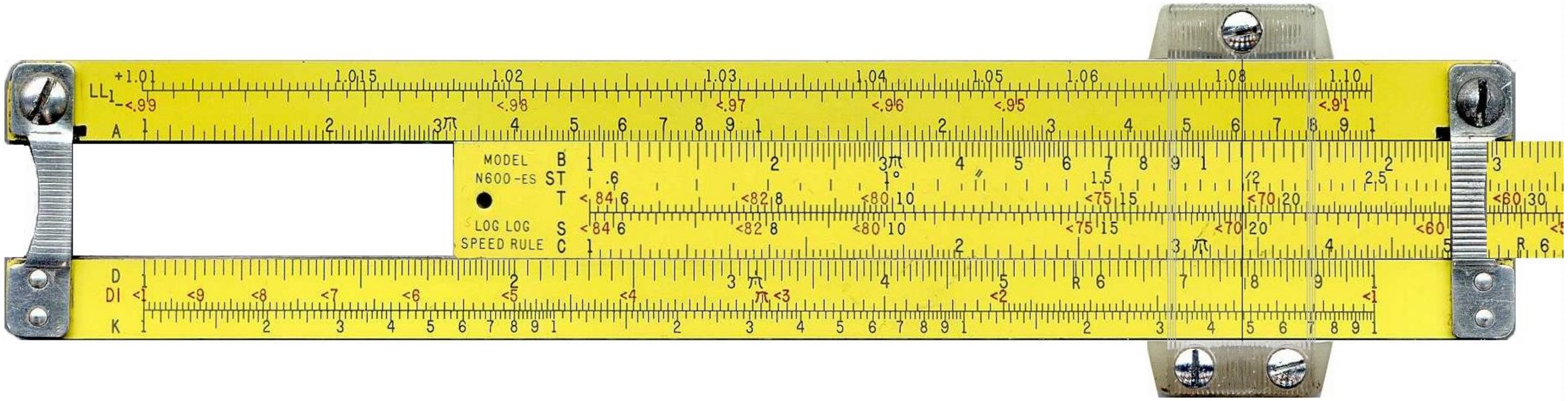
- Robot collects data
- Models are learned in the cloud
- Regression tests in the cloud
- Failures are sent to the annotators
- The annotated data becomes part of the training set



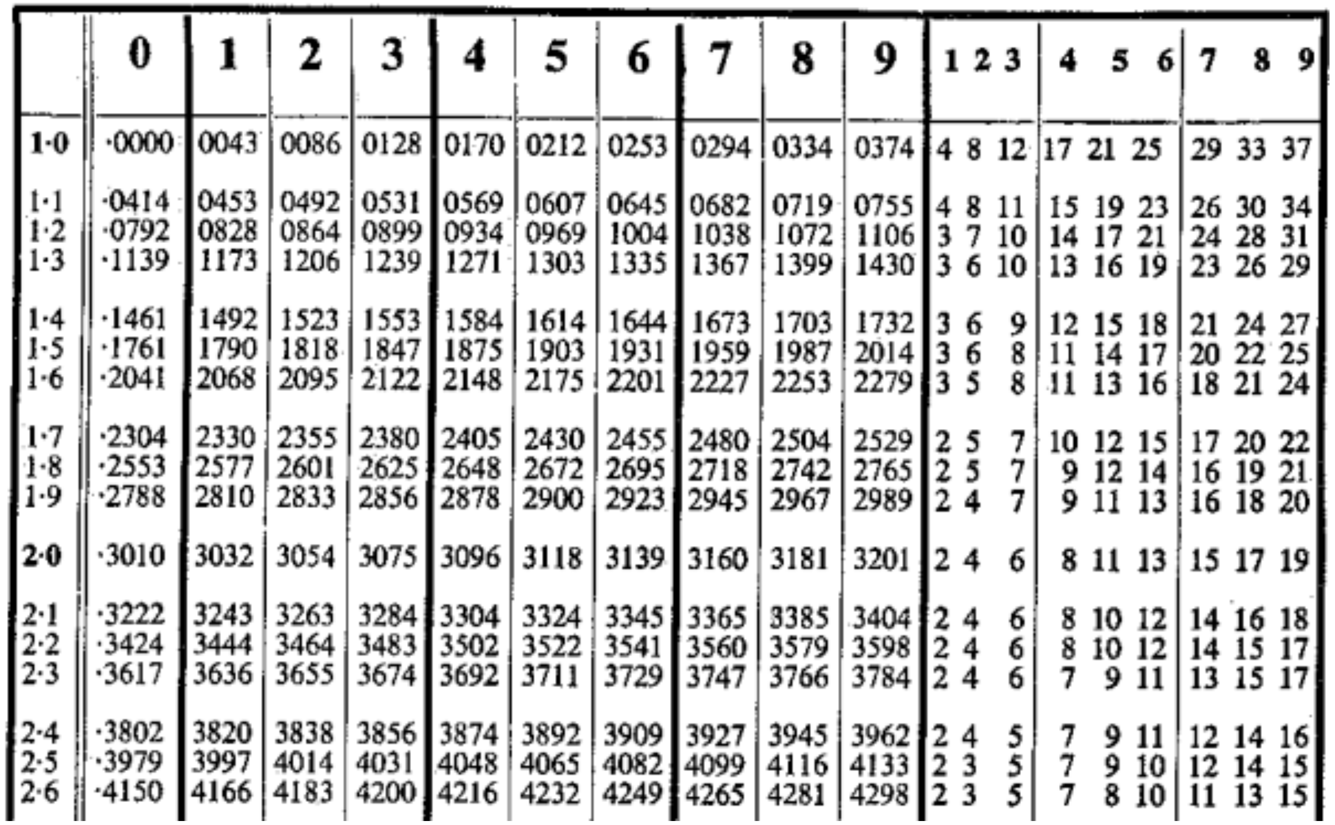
Tools for modern roboticists

Tools make the professional

Engineering tools of the 40s



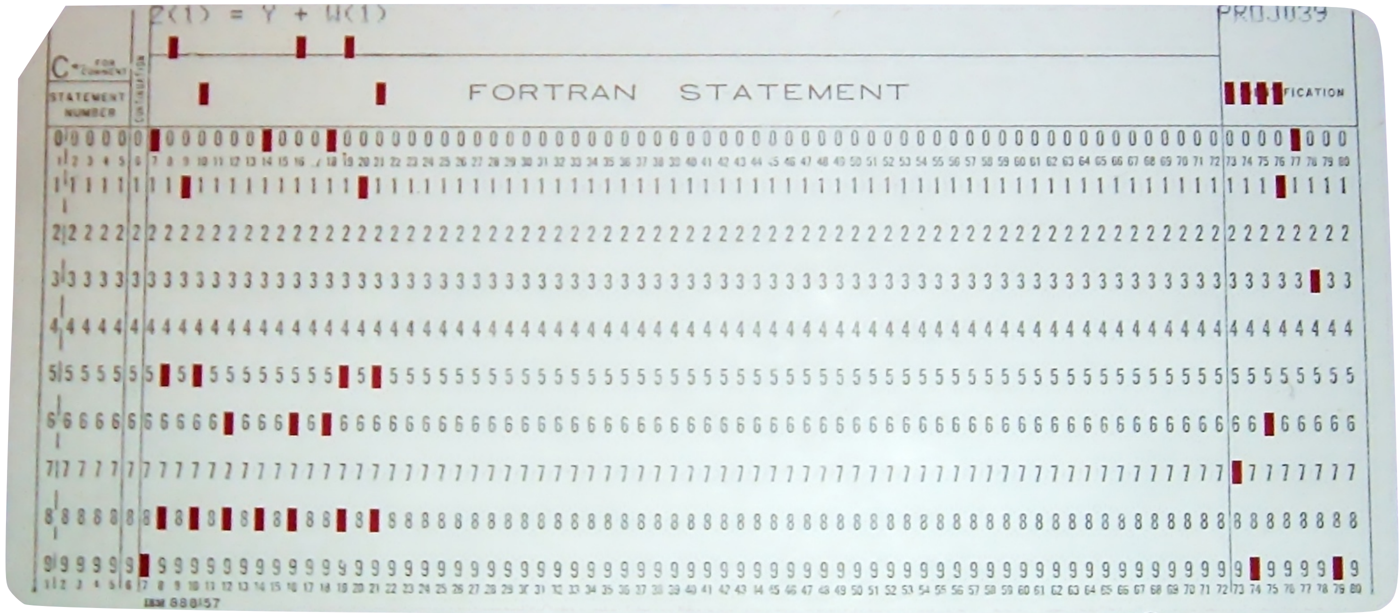
Slide rule



	0	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
1·0	·0000	0043	0086	0128	0170	0212	0253	0294	0334	0374	4	8	12	17	21	25	29	33	37
1·1	·0414	0453	0492	0531	0569	0607	0645	0682	0719	0755	4	8	11	15	19	23	26	30	34
1·2	·0792	0828	0864	0899	0934	0969	1004	1038	1072	1106	3	7	10	14	17	21	24	28	31
1·3	·1139	1173	1206	1239	1271	1303	1335	1367	1399	1430	3	6	10	13	16	19	23	26	29
1·4	·1461	1492	1523	1553	1584	1614	1644	1673	1703	1732	3	6	9	12	15	18	21	24	27
1·5	·1761	1790	1818	1847	1875	1903	1931	1959	1987	2014	3	6	8	11	14	17	20	22	25
1·6	·2041	2068	2095	2122	2148	2175	2201	2227	2253	2279	3	5	8	11	13	16	18	21	24
1·7	·2304	2330	2355	2380	2405	2430	2455	2480	2504	2529	2	5	7	10	12	15	17	20	22
1·8	·2553	2577	2601	2625	2648	2672	2695	2718	2742	2765	2	5	7	9	12	14	16	19	21
1·9	·2788	2810	2833	2856	2878	2900	2923	2945	2967	2989	2	4	7	9	11	13	16	18	20
2·0	·3010	3032	3054	3075	3096	3118	3139	3160	3181	3201	2	4	6	8	11	13	15	17	19
2·1	·3222	3243	3263	3284	3304	3324	3345	3365	3385	3404	2	4	6	8	10	12	14	16	18
2·2	·3424	3444	3464	3483	3502	3522	3541	3560	3579	3598	2	4	6	8	10	12	14	15	17
2·3	·3617	3636	3655	3674	3692	3711	3729	3747	3766	3784	2	4	6	7	9	11	13	15	17
2·4	·3802	3820	3838	3856	3874	3892	3909	3927	3945	3962	2	4	5	7	9	11	12	14	16
2·5	·3979	3997	4014	4031	4048	4065	4082	4099	4116	4133	2	3	5	7	9	10	12	14	15
2·6	·4150	4166	4183	4200	4216	4232	4249	4265	4281	4298	2	3	5	7	8	10	11	13	15

Log table

Tools of the 1950s

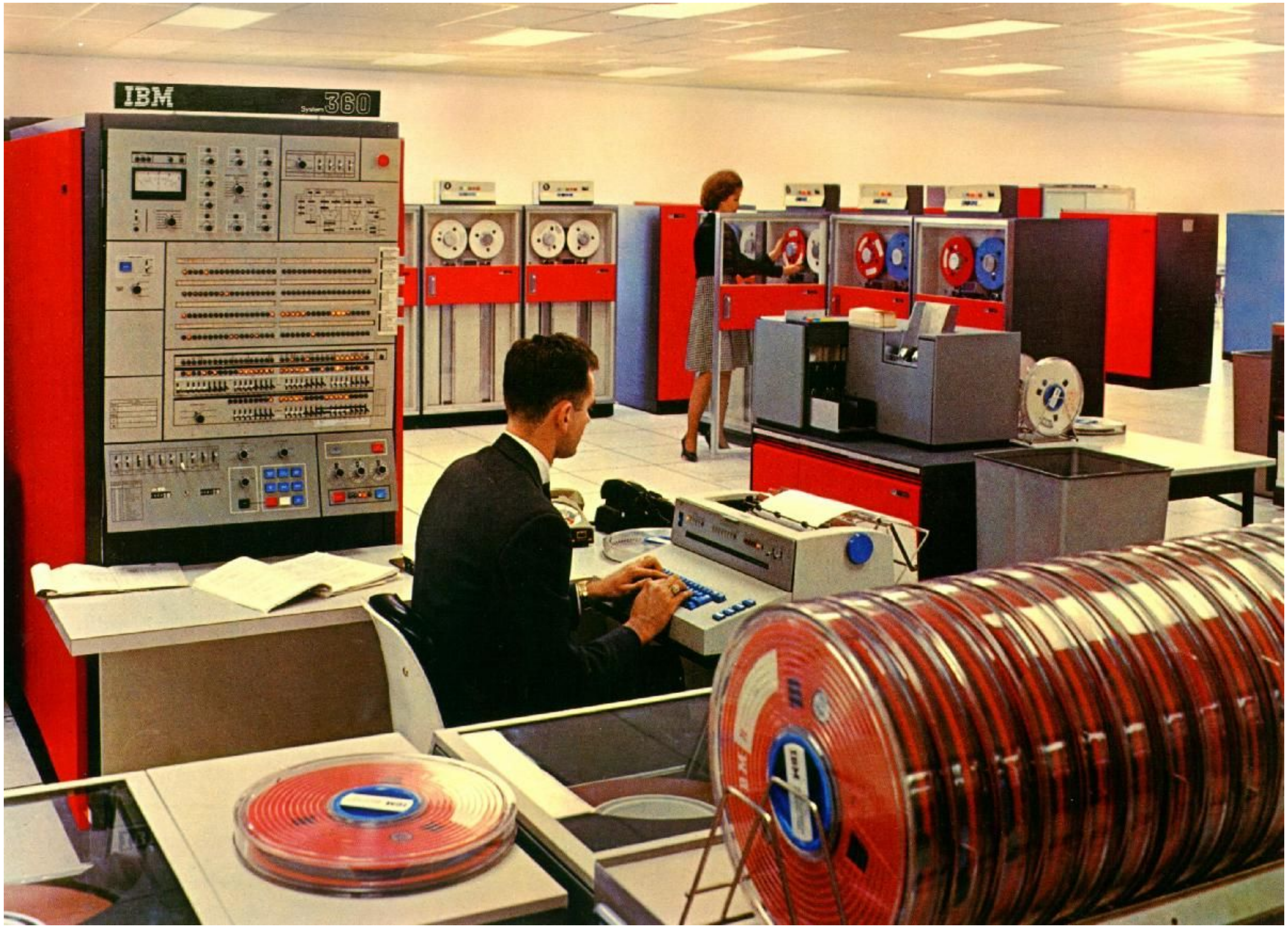


Punched card

This stack of 62,500 punched cards — 5 MB worth — held the control program for the giant SAGE military computer network. (link)

Tools of the 1960s

Mainframe



Tools of the 1960s

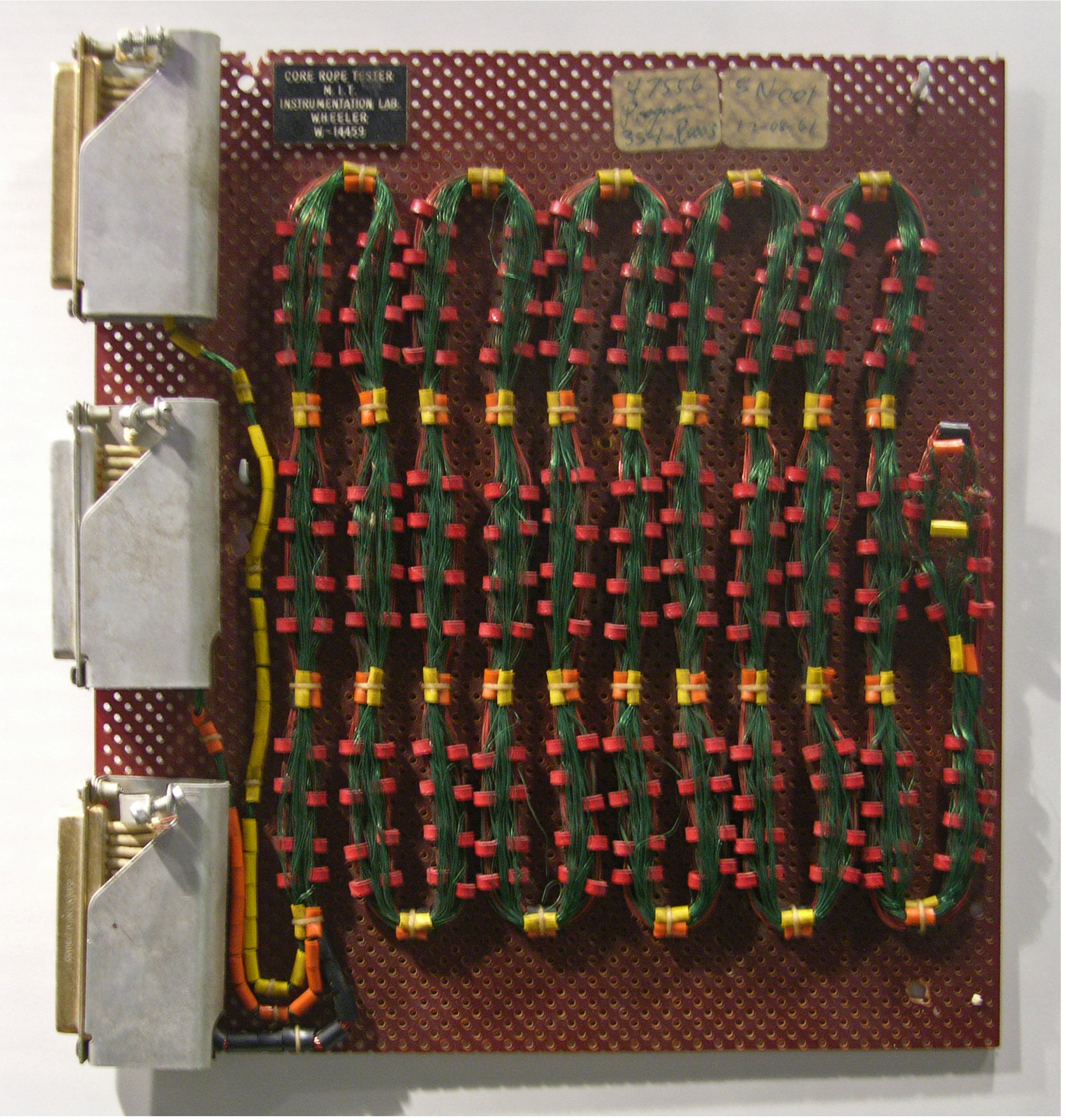
- Director of the Software Engineering Division of the MIT Instrumentation Laboratory next to the **source code for Apollo 11**.

    - You can find the code on Github at: https://github.com/chrislgarry/Apollo-11

Margaret Hamilton (1936-)

“Core rope memory” used by the Apollo program

- Computer memory was represented by magnetic ropes, woven together by old ladies.
  - LOL memory = **Little Old Lady memory**
- One of the main problems was reducing the source code size, and the slow speed of old ladies.
  - Read about this and other interesting stories in the Tindellgrams.

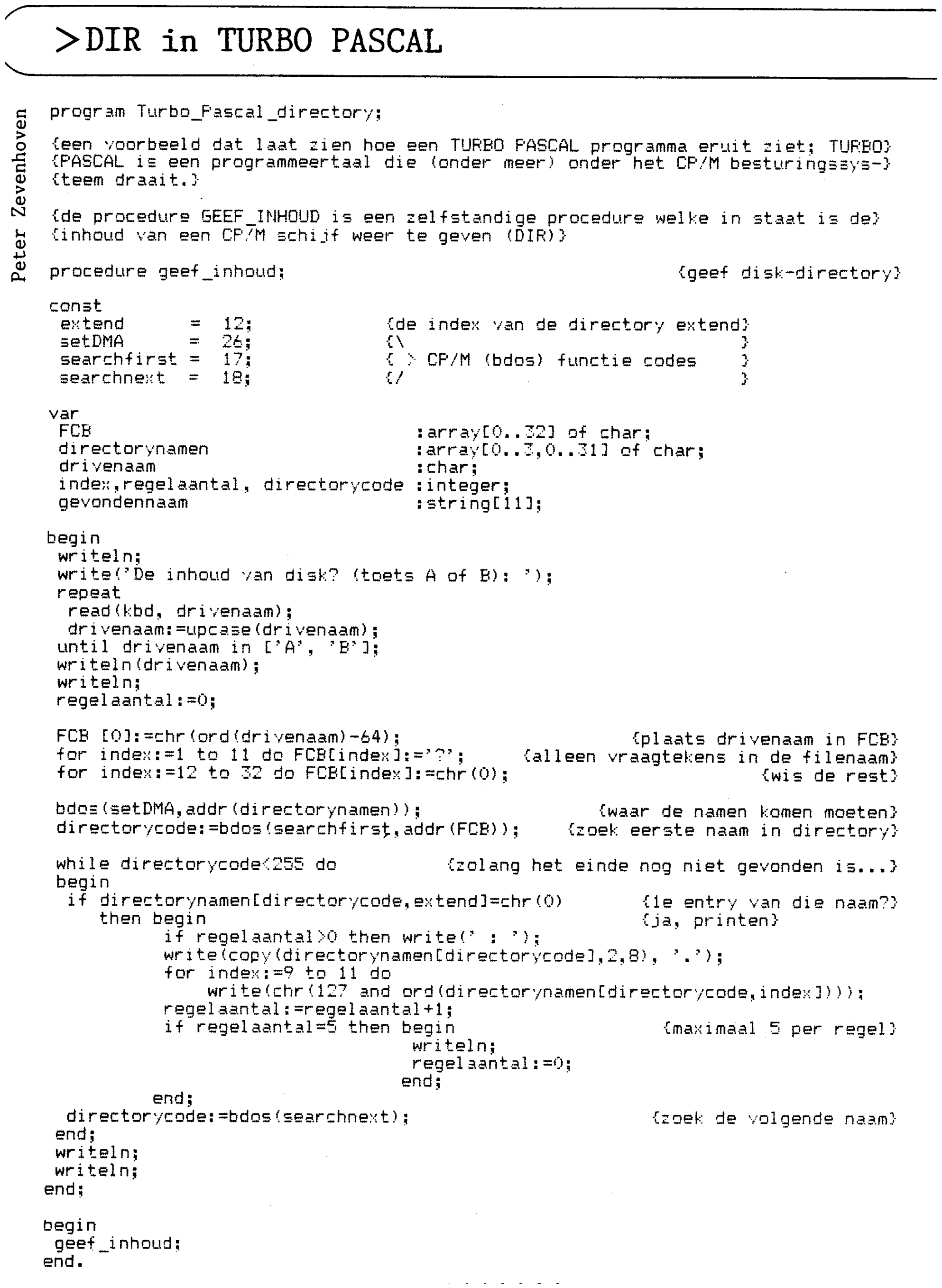


Rope memory

Tools of the 1970s

HP-35

Pascal



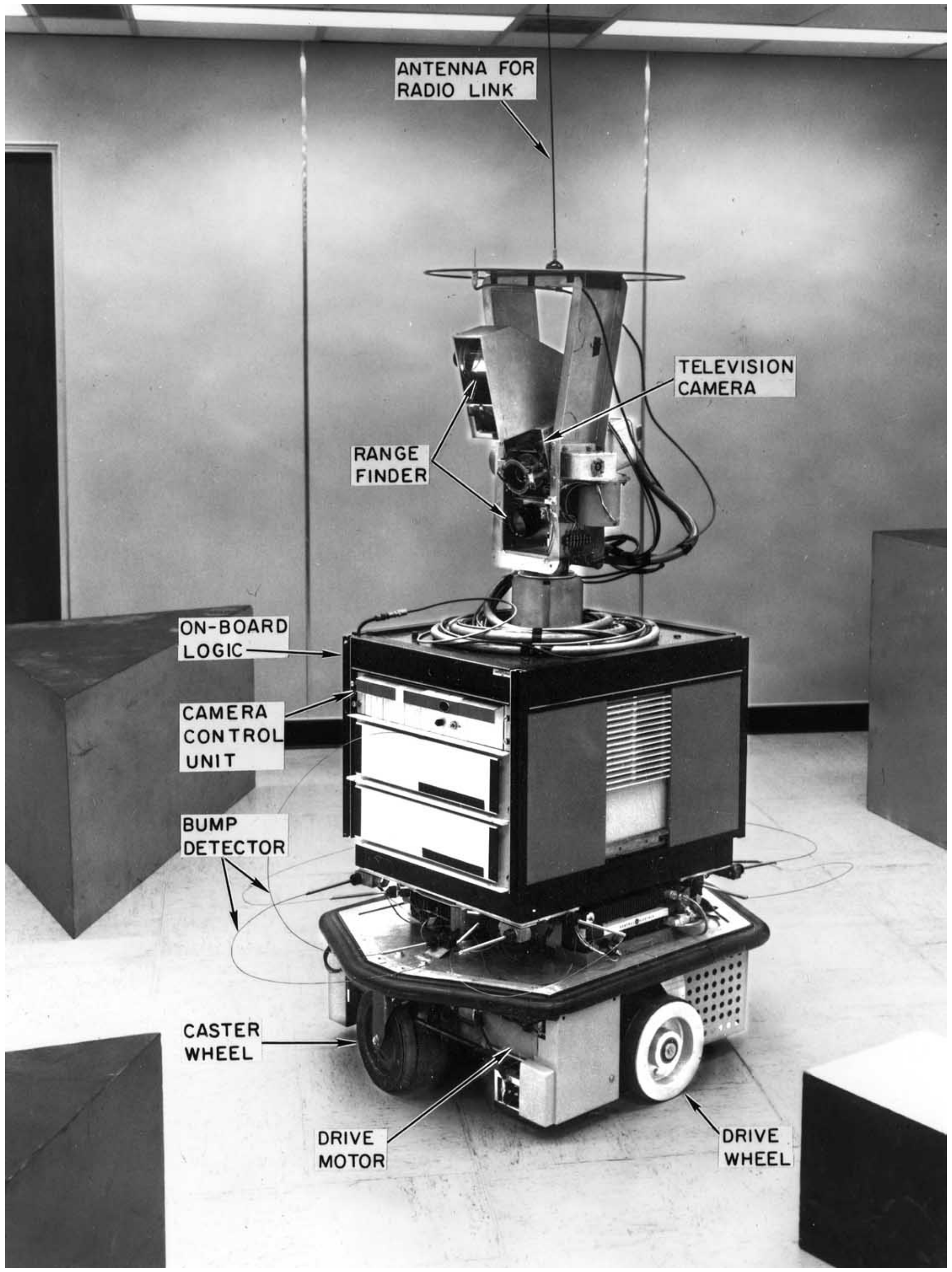
>DIR in TURBO PASCAL

Peter Zevenhoven

```
program Turbo_Pascal_directory;

{een voorbeeld dat laat zien hoe een TURBO PASCAL programma eruit ziet; TURBO}
{PASCAL is een programmeertaal die (onder meer) onder het CP/M besturingssys-}
{teem draait.}

{de procedure GEEF_INHOUD is een zelfstandige procedure welke in staat is de}
{inhoud van een CP/M schijf weer te geven (DIR)}

procedure geef_inhoud;                                    {geef disk-directory}

const
 extend      =  12;              {de index van de directory extend}
 setDMA      =  26;              {\                               }
 searchfirst =  17;              { > CP/M (bdos) functie codes    }
 searchnext  =  18;              {/                               }

var
 FCB                                 :array[0..32] of char;
 directorynamen                      :array[0..3,0..31] of char;
 drivenaam                           :char;
 index,regelaantal, directorycode    :integer;
 gevondennaam                        :string[11];

begin
 writeln;
 write('De inhoud van disk? (toets A of B): ');
 repeat
  read(kbd, drivenaam);
  drivenaam:=upcase(drivenaam);
 until drivenaam in ['A', 'B'];
 writeln(drivenaam);
 writeln;
 regelaantal:=0;

 FCB [0]:=chr(ord(drivenaam)-64);                       {plaats drivenaam in FCB}
 for index:=1 to 11 do FCB[index]:='?';     {alleen vraagtekens in de filenaam}
 for index:=12 to 32 do FCB[index]:=chr(0);                        {wis de rest}

 bdos(setDMA,addr(directorynamen));                 {waar de namen komen moeten}
 directorycode:=bdos(searchfirst,addr(FCB));    {zoek eerste naam in directory}

 while directorycode<255 do          {zolang het einde nog niet gevonden is...}
 begin
  if directorynamen[directorycode,extend]=chr(0)         {1e entry van die naam?}
     then begin                                          {ja, printen}
           if regelaantal>0 then write(' : ');
           write(copy(directorynamen[directorycode],2,8), '.');
           for index:=9 to 11 do
               write(chr(127 and ord(directorynamen[directorycode,index])));
           regelaantal:=regelaantal+1;
           if regelaantal=5 then begin                   {maximaal 5 per regel}
                                  writeln;
                                  regelaantal:=0;
                                 end;
          end;
  directorycode:=bdos(searchnext);                      {zoek de volgende naam}
 end;
 writeln;
 writeln;
end;

begin
 geef_inhoud;
end.
```

Robots of the 1970s

- “Shakey”, the first “embodied intelligence”.
  - Documentary.



Shakey the robot

Tools of the 1980s

Amstrad 64K Microcomputer (v1)
©1984 Amstrad Consumer Electronics plc
and Locomotive Software Ltd.
BASIC 1.0
Ready
AMSTRAD

Personal computers

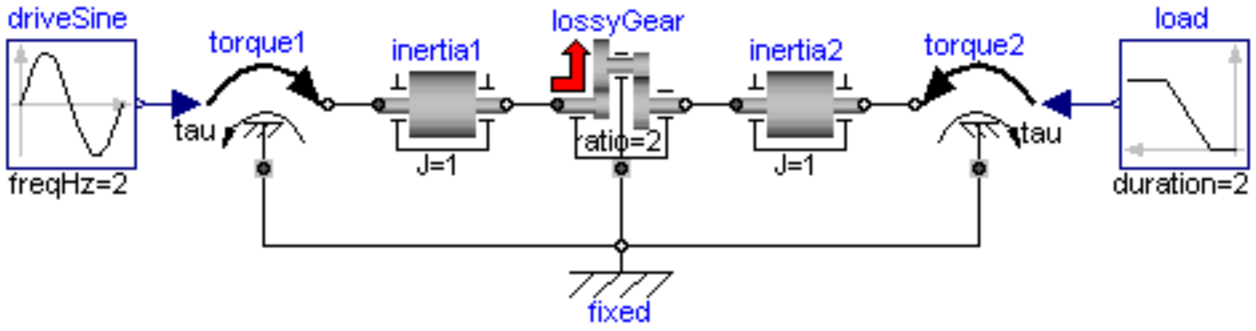
Tools of the 1990s

- Modelling software
- Computer-algebra systems

Wolfram Mathematica

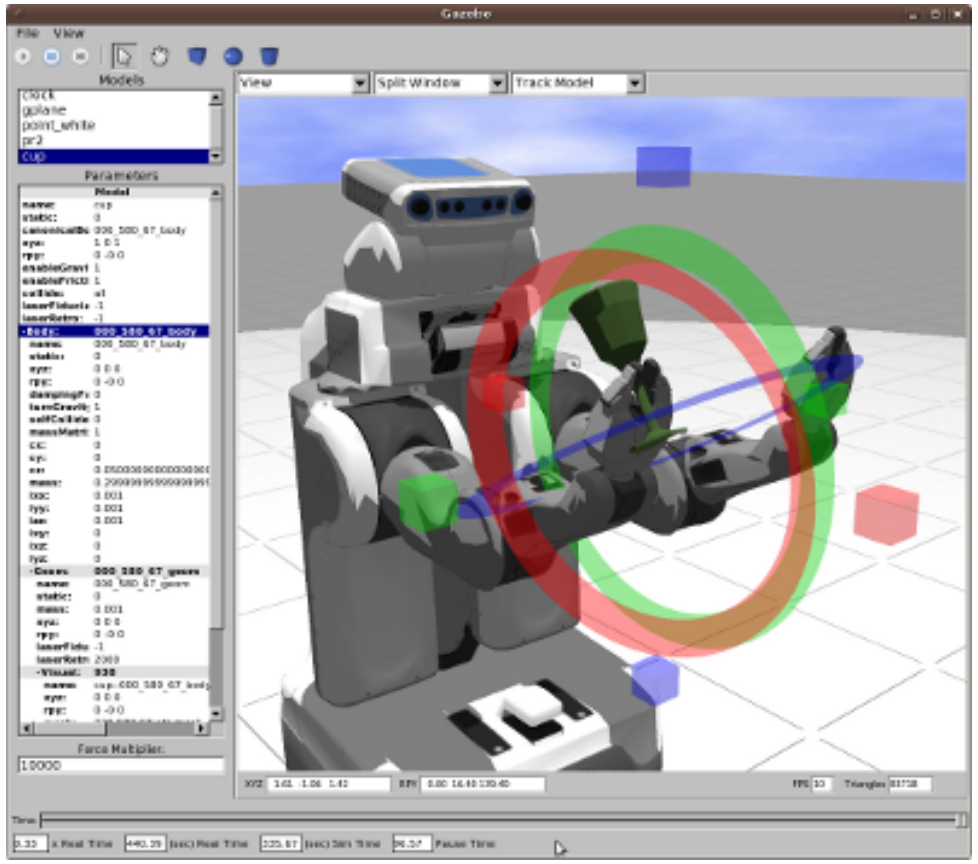
MATLAB

MODELICA

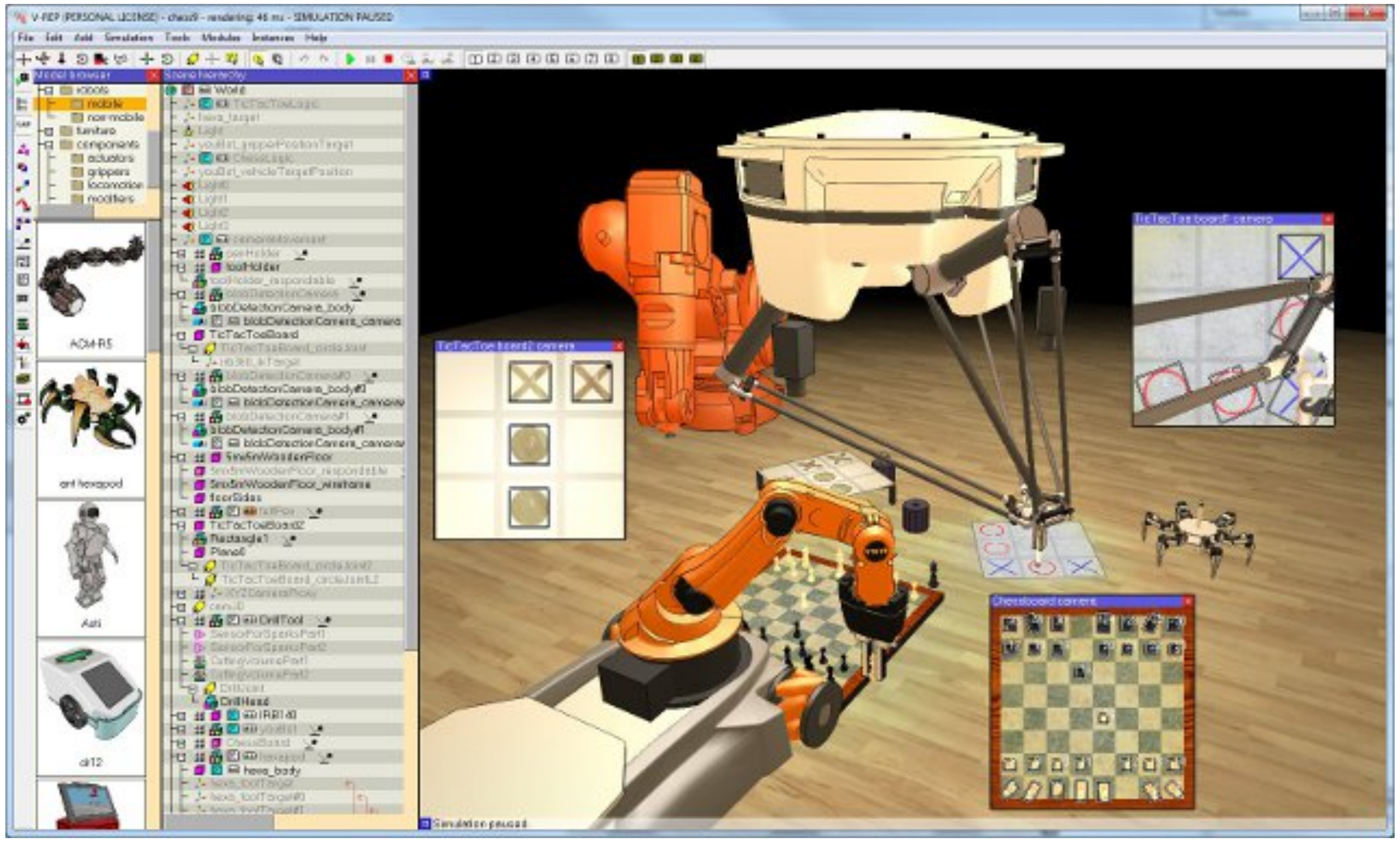


Tools of the 2000s

- Robot simulations



Gazebo



V-Rep

Tools of the 2010s

- Source code management
- Issue tracking
- Cloud storage
- Cloud computing
- Continuous integration
- Configuration management
- Project management

What's cool in 2017

still cool in 2018!

and 2019!

and 2020!

git

GitHub

circleci

amazon web services

docker

new entry

Take-away points

- The robot itself is just a small part of a **modern robotic system**, which includes robots, people, local infrastructure, remote infrastructure.

- There are new **development methods and tools** to learn.
  - Formulas are useless if you cannot translate them into action.
  - Nobody will do it for you!

“smart, and gets things done”

- p.s. You are not competing with whom is around you, *you are competing with everybody else in the world…*