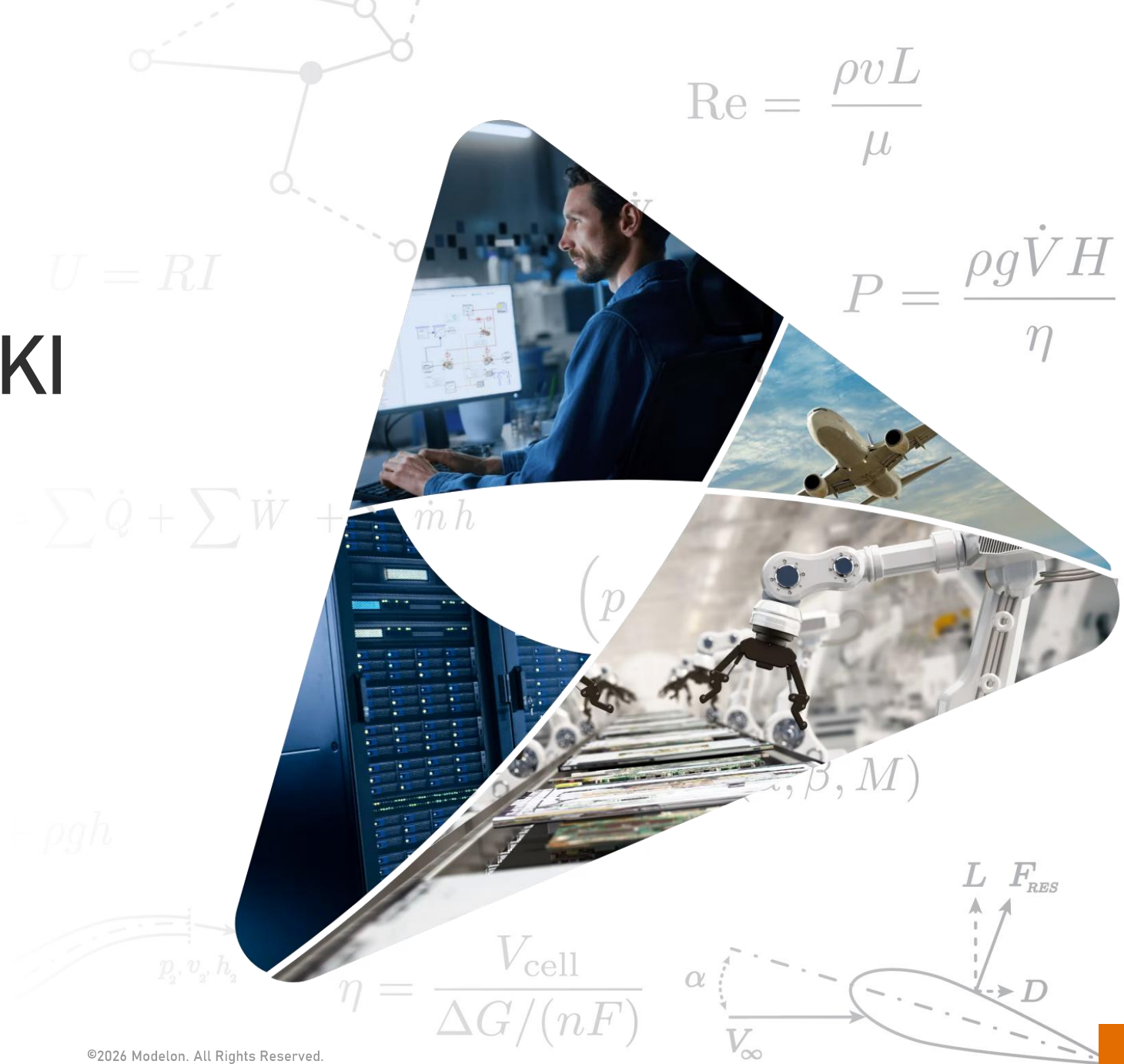


Einsatz generativer KI zur Erstellung von Systemmodellen

Daniel Rohde & Eberhard Erb

Modelon Deutschland GmbH

23.09.26



Modelon

- Schwedisches Unternehmen
- Angebot
 - Cloud-Plattform für Systemsimulation
 - Mehrere Modelica-Bibliotheken
 - Simulationsdienstleistungen



LIBRARIES

- ▶  AirConditioning 3.0
- ▶  DataCenter 2.0
- ▶  Electrification 2.2
- ▶  EnergySystems 2.0
- ▶  FuelCell 3.0
- ▶  HeatExchanger 5.0
- ▶  Hydraulics 4.25
- ▶  LiquidCooling 4.0
- ▶  Modelon 7.0
- ▶  ThermalPower 4.0
- ▶  VaporCycle 5.0
- ▶  VehicleDynamics 4.9

Keynote auf der Asian Modelica Conference (21.9.)

Dr. Dirk Zimmer: “The Rise of Computer Intelligence and what it means for System Simulation“

一些语言将作为计算框架变得不再那么重要,或者越来越不重要。但随着它们成为我们与计算机智能之间的通信工具,它们将变得更加相关。我认为实际上,Modelica和FMI都非常适合。我认为我们确实能够履行这一职责。我们并没有为此设计,但我们会尽力做主要正确的事情,我认为这现在真的帮到了我们。因此我们希望语言具有陈述性。实际上它并没有说明如何进行计算。因此我们可以采取非常不同的方法。计算机智能可以处理内容并说“好的,我会以这种方式处理它,因为这对这个模型来说有意义。”“所以最初,如果你回看15或20年之前,它某种程度上与某个计算框架相关联。所以基本上我们只有一种主导工具,而且它必须以某种方式进行处理。现在15年后,我们确实努力使其成为一个开放标准。因此,你有许多支持建模的工具,还有更多即将出现。因此我们有RumoCarl以及SimTech的新工具,这些工具可以用于建模。我看到越来越多的工具出现。主要重要的是,我们确实认真对待了这件事,通过开放的社区开发了一个开放标准。我认为这确实是一个关键点。

And this is also something that is also very important for the FMI.

What will be the role of Modelica (& FMI) w.r.t. AI?

- Modelica (and also FMI) is luckily well suited for such a transition.
- Modelica is principally designed to be declarative and oblivious to its computational realization.
- Whereas initially it was practically bound to a dominating computational framework, the work of the last 15 years, really turned it into a standard.
- We represent an open standard with an open community.
- This also applies to FMI
- The times of language-based vendor lock-in are definitely over with AI (already today).
- The communication layer is what will matter for humans

DLR

Machine Code + Binary Intermediate Representations

fmi modelica Language

Time (billion years before present)

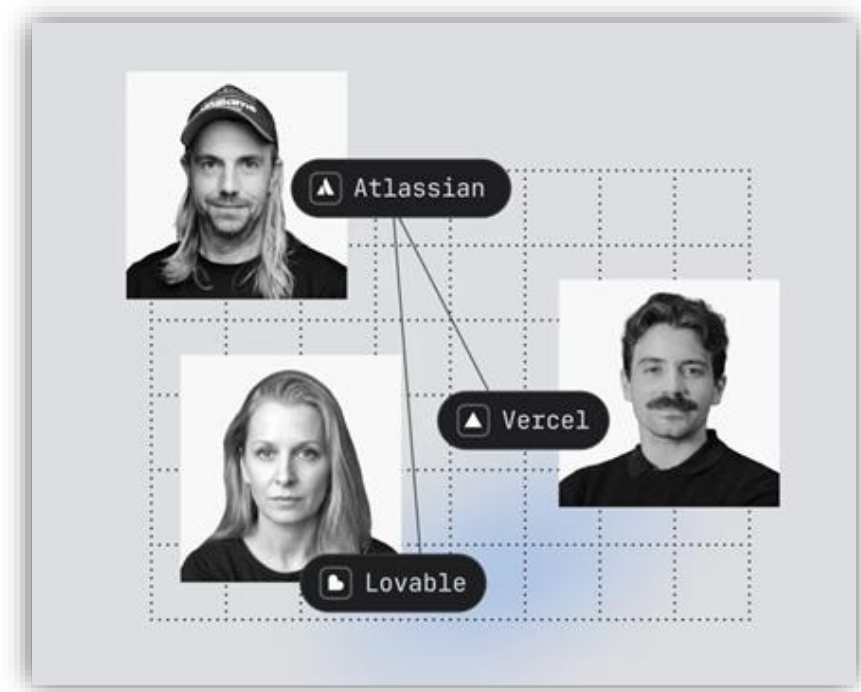
-2.5 -2.0 -1.5 -1.0 -0.5 0.0

future

15

Dirk Zimmer, Institute of Robotics and Mechatronics

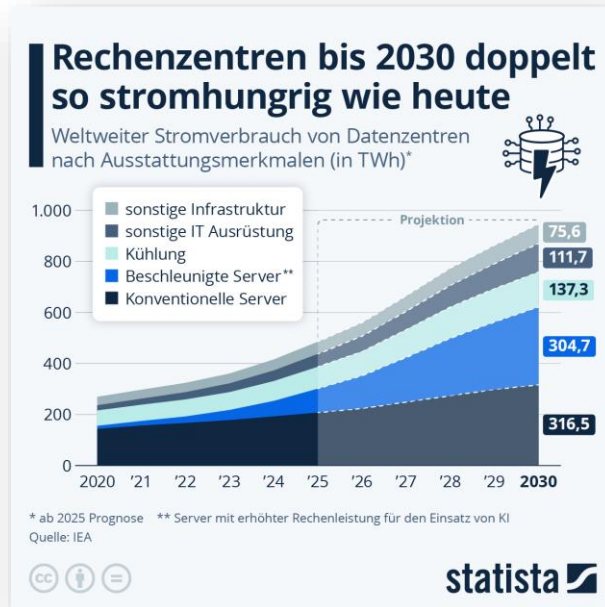
Atlassian Digital Summit (22.9.)



Schon genervt?



Oder sogar besorgt?



Evan Hubinger ✓
@EvanHub

Jacob is correct here—we really do earnestly believe AI could kill all humans! I personally think it is >10% within the next decade. I believe Anthropic is trying its best, but we do not yet have a plan to solve alignment for superintelligence and are not clearly on track to.



Frankfurter Allgemeine



Wirtschaft > Unternehmen > Amodi, Altman, Mu:

KÜNSTLICHE INTELLIGENZ

Bringt die KI uns alle um?

Von Patrick Bernau, Alexander Wulfers

13.09.2026, 09:49 Lesezeit: 7 Min.



Gruseln wie im Film: James Camerons Terminator Picture Alliance

KI-Entwickler warnen davor, dass ihre Schöpfung die Menschheit vernichten kann. Nun verlangen sogar die Chefs von Open AI und Anthropic öffentlich, die Forschung zu bremsen – und Elon Musk stimmt ihnen zu. Wie ernst muss man die Sorgen nehmen?

DER SPIEGEL

Anmelden



Ihr tägliches Extra bei SPIEGEL+



WORK

Hilfe, die KI klagt mir meinen Job

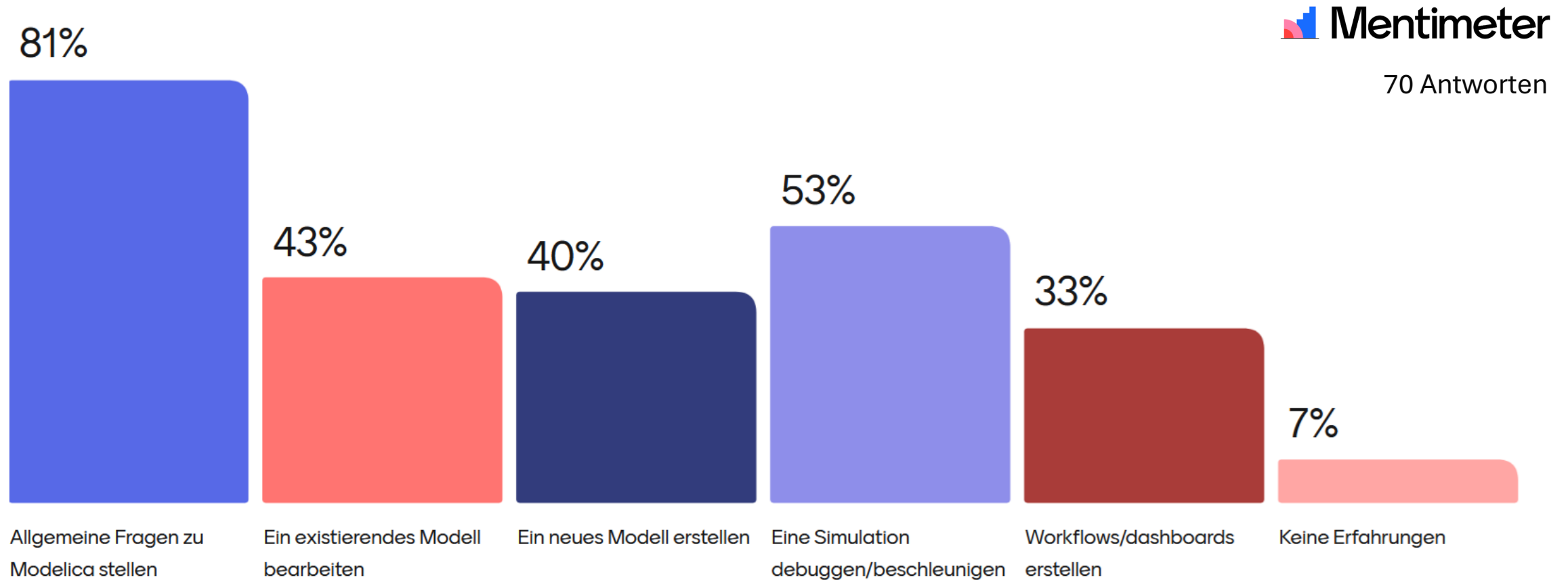
KI krempelt die Arbeitswelt um. Besonders in administrativen Bereichen haben viele Arbeitnehmer Angst, abgehängt zu werden – oder gleich ganz überflüssig. Unsere Coachin erklärt, wie Sie sich zukunftsfit aufstellen.

Von Doris Brenner

Kurze Umfrage zum Thema KI & Modelica



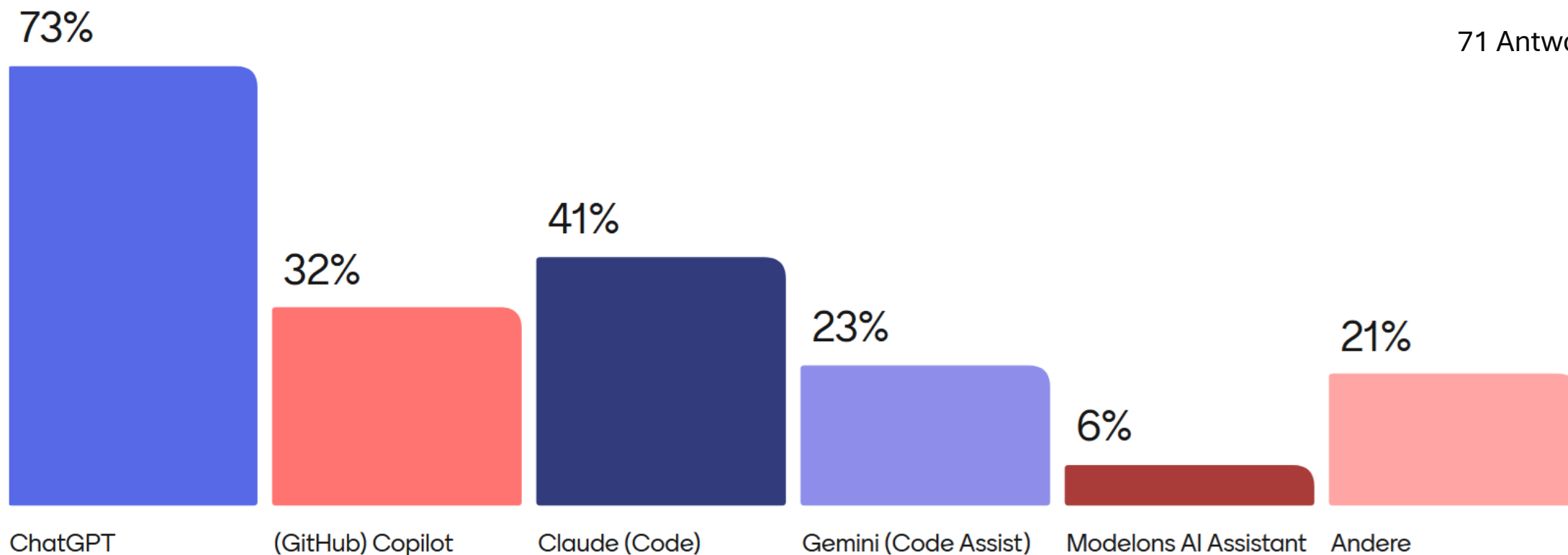
Dafür habe ich KI bereits (erfolgreich) eingesetzt:



Verwendet und/oder getestet habe ich:

 **Mentimeter**

71 Antworten



Diese Begriffe fallen mir zu KI & Modelica ein:



51 Antworten



KI-Umfrage von McKinsey (08/2026)



“Organizations are deploying agentic coding tools and coming to grips with the costs of AI [...].”

“AI use is boosting the performance of individuals at work. 80 % of respondents report that AI has **improved their individual productivity**, and 50 % report that AI **helps them make better decisions.**”

Generative KI sinnvoll nutzen - wie?

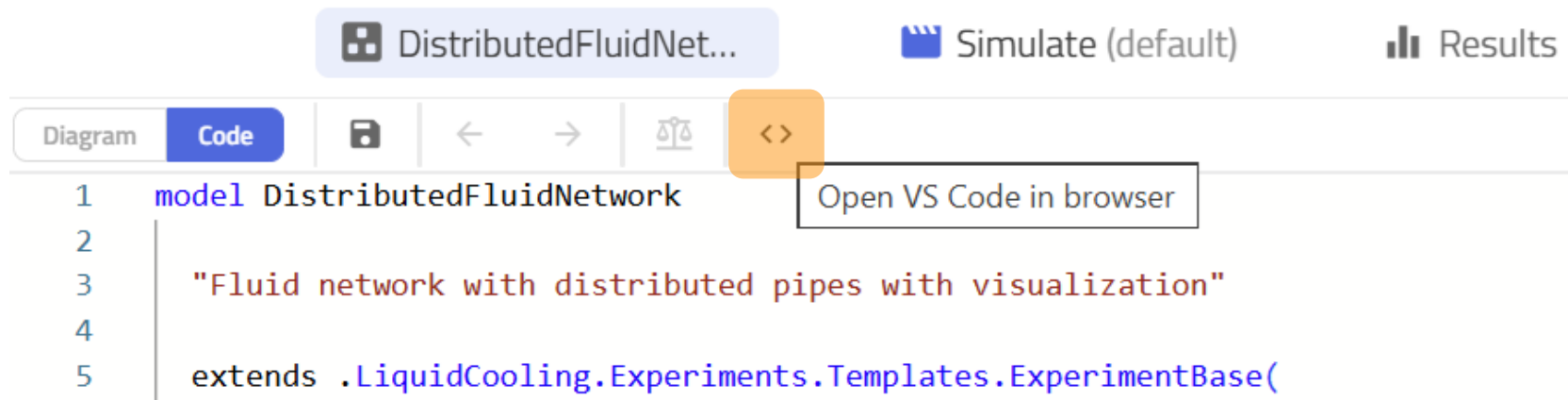
Tools müssen reagieren

Nutzer müssen sich umstellen

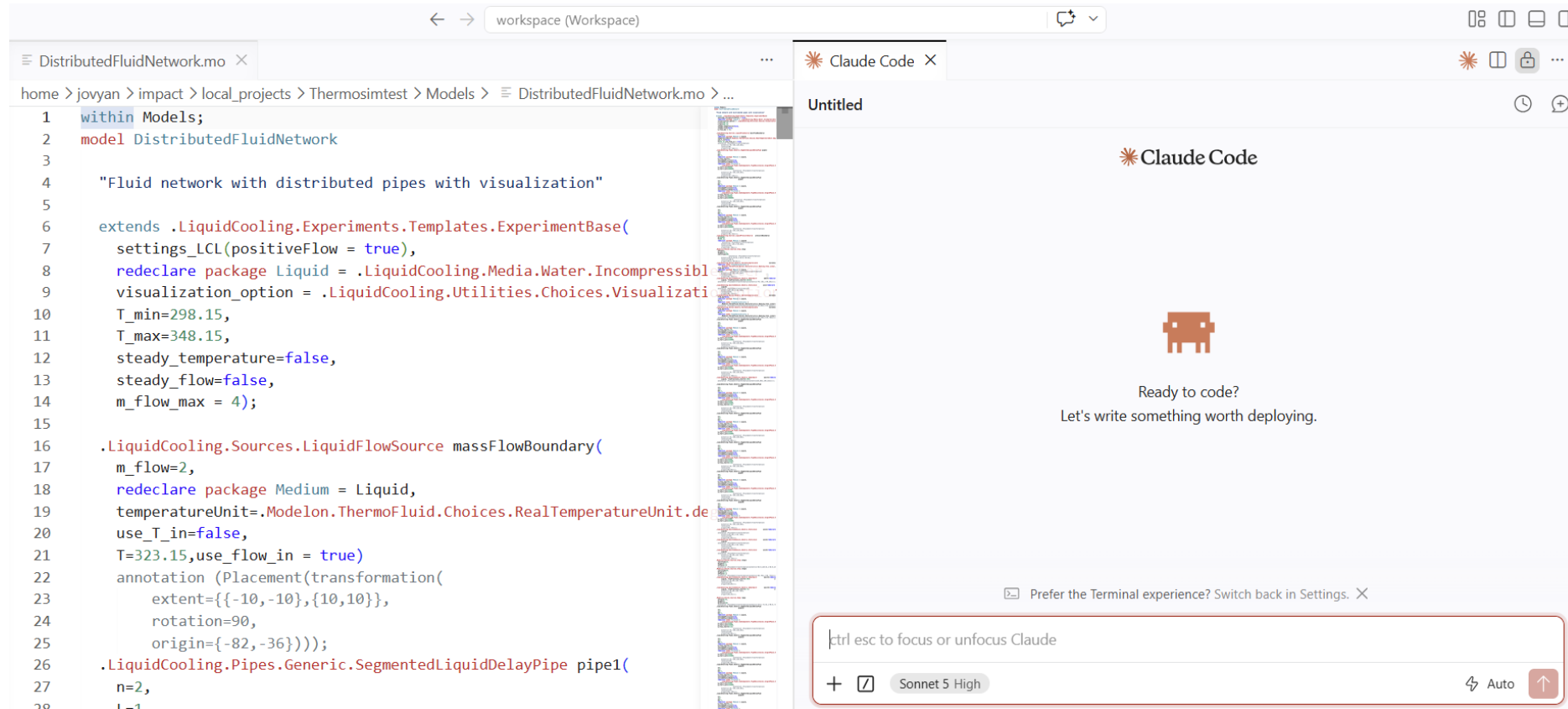
“A fool with a tool is still a fool”

Einblicke in Modelons “KI-Reise”

Generativ zuerst über Kopplung mit Claude Code (VS Code in Browser)



Einblicke in Modelons “KI-Reise”

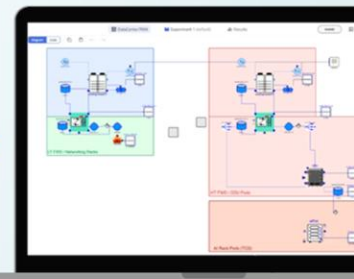


Einblicke in Modelons “KI-Reise”

From Spec to Simulation: An AI Agent Builds a Data Center Cooling Model

From Spec to Simulation

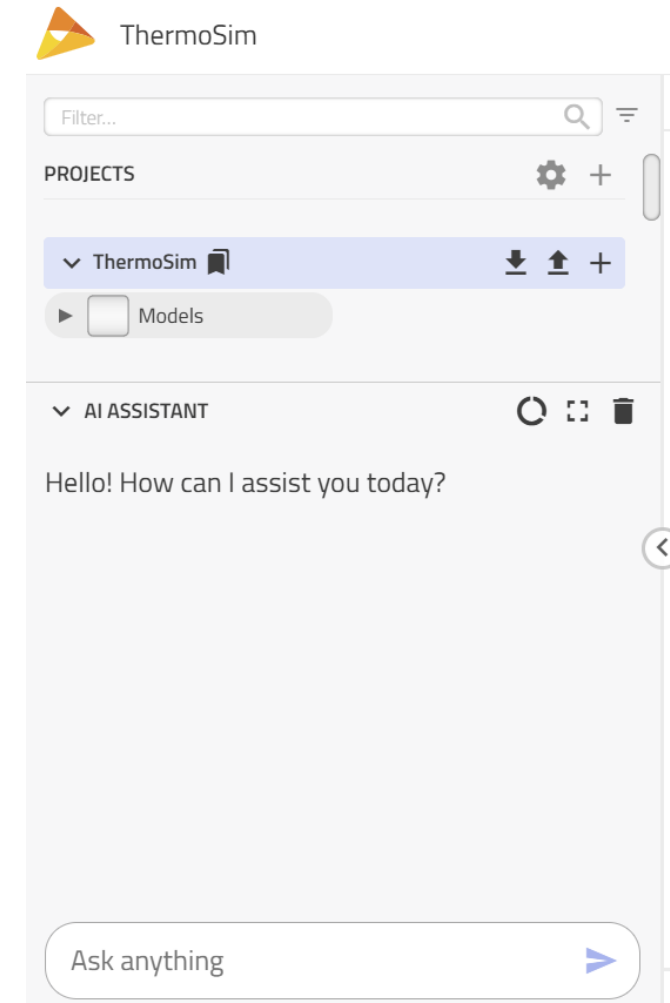
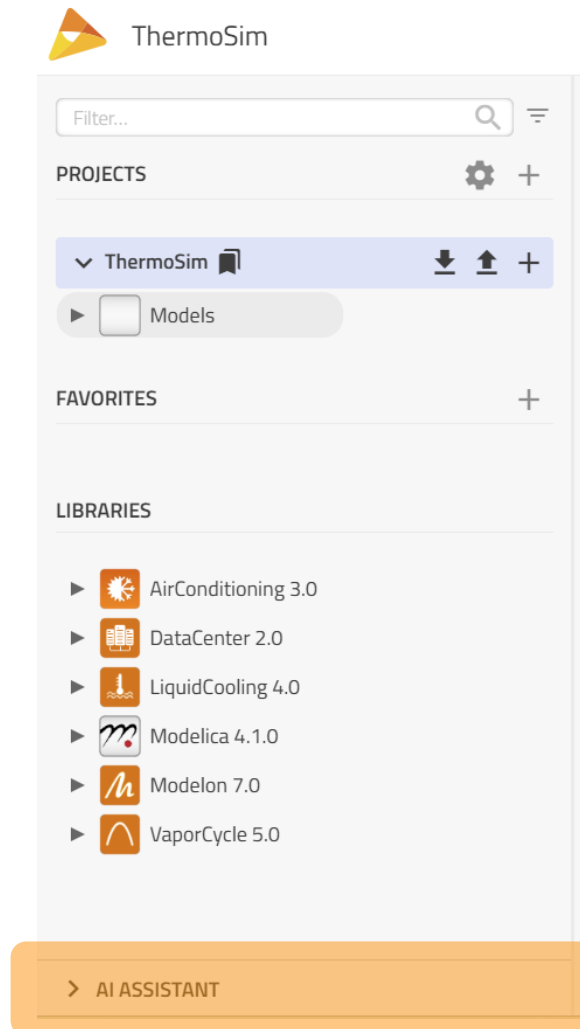
Modelon Impact



Nithish Selvan | 06/10/2026

Einblicke in Modelons “KI-Reise”

- Entwicklung des “AI Assistant”
- Nutzerfreundlich integriert in Impact GUI
- Erste “Demo” auf Modelica-Konferenz 2025



Einblicke in Modelons “KI-Reise”

Aktuell (Beta)

- Onboarding: Fragen zum Tool, Modelica, Bibliotheken, etc.
- Debugging
- Analyse von Ergebnissen
- Numerisches “tuning”

Bald (Alpha)

- Deutlich bessere Qualität
- „Agentic“: Modelle erstellen und ändern

Später

- Versionskontrolle?
- Apps/Workflows?
- Kundenspezifisches Training?

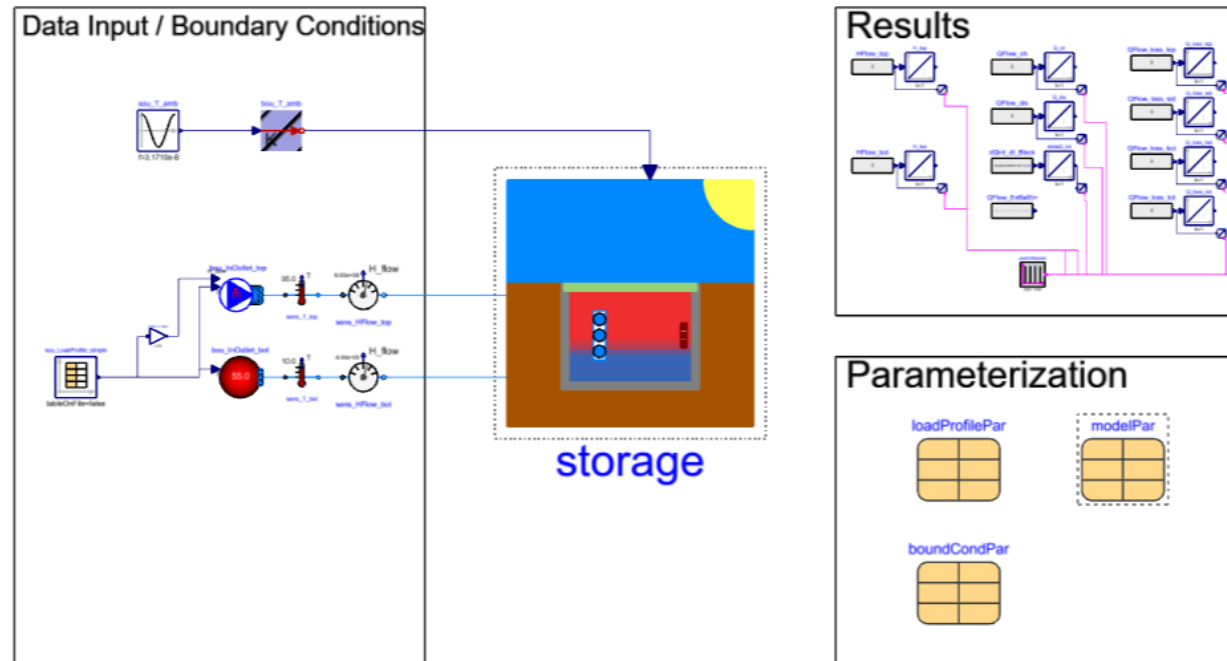
“Modelon’s AI Assistant proved highly valuable for troubleshooting, as it helped us to quickly identify where we had made mistakes, understand the simulation log, and find clear steps to resolve the issues. It also supported proper component parametrization, making the modelling process faster, more transparent, and easier to navigate.”

Jan Škruba
Project Engineer at Resalta

Kleines Beispiel (Alpha)

<https://github.com/LargeTESmtk/LargeTESmtk>

LargeTESmtk.TankTES.Cylinder.Examples.FullyBuried_Basic.UniformGround



Kleines Beispiel (Alpha)

AI ASSISTANT

Hello! How can I assist you today?

LargeTESmtk.TankTES.Cylinder.Examples.FullyBuried_Basic.UniformGround kompiliert nicht. Analysiere den Fehler und schlage eine Lösung vor.

✓ Loading skill instructions 0.0s

✓ Checking model class type 2.5s

✓ Fetching simulation IDs 1.7s

✓ Fetching simulation log 0.4s

✓ Fetching model source 5 calls 0.9s

Empfehlung

Der Fehler liegt in der Basisklasse `LargeTESmtk.BaseClasses.TESModels.PartialXTES_Cylinder_FullyBuried_CustomGD` (verwendet über die Komponente `storage` im Beispielmmodell). Zeile 52 definiert:

```
parameter Modelica.Media.Interfaces.Types.AbsolutePressure p_start = fluidDom.Medium.p_default
```

Das ist ungültig: `Medium` ist selbst eine Klasse (ein austauschbares Package), kein normaler Wert — und der Zugriff auf eine Nicht-Funktions-Klasse über eine Komponenteninstanz (`fluidDom.Medium...`) ist in Modelica nicht erlaubt ("Cannot access non-function class through component access").

Lösung: Den Ausdruck so ändern, dass er direkt auf das im Modell selbst deklarierte, austauschbare `Medium`-Package `storageMedium` zugreift statt über die Instanz `fluidDom`:

```
parameter Modelica.Media.Interfaces.Types.AbsolutePressure p_start = storageMedium.p_default
```

Diese Schreibweise wird an anderer Stelle in der Bibliothek bereits korrekt so verwendet (z. B. in `PartialFluidDomain`: `parameter Medium.AbsolutePressure p_start = Medium.p_default`), dort allerdings im eigenen Scope, nicht über eine Komponente.

Kleines Beispiel (Alpha)

AI ASSISTANT

Erklärung

fluidDom ist eine Instanz von `LargeTESmtk.Components.FluidDomain.Cylinder`, in der `storageMedium` als austauschbares Package redefiniert wird. Der Ausdruck `fluidDom.Medium.p_default` versucht, über diese Instanz auf die Medium-Klasse selbst zuzugreifen — das ist laut Modelica-Sprachspezifikation nur für Funktionen erlaubt, nicht für beliebige Klassen wie Packages. Da `storageMedium` im umgebenden Modell (`PartialXTES_Cylinder_FullyBuried_CustomGD`) bereits als eigenes Parameter-Package deklariert ist, kann direkt darauf zugegriffen werden, ohne den Umweg über die Komponenteninstanz.

Möchten Sie, dass ich diese Codeänderung im Quellcode für Sie vornehme?

ja

✓ Loading skill instructions 0.4s

✓ Running script 7.2s

Datei und Stelle gefunden. Ich bestätige die Änderung, bevor ich sie schreibe.

Ich ändere in `PartialXTES_Cylinder_FullyBuried_CustomGD.mo` (Projekt `LargeTESmtk`) die Zeile: ``parameter Modelica.Media.Interfaces.Types.AbsolutePressure p_start = fluidDom.Medium.p_default`` zu: ``parameter Modelica.Media.Interfaces.Types.AbsolutePressure p_start = storageMedium.p_default`` Soll ich das jetzt schreiben?

Ja, schreiben

✓ Running script 3.0s

✓ Running script 0.4s

Die Änderung ist geschrieben: `p_start` bezieht sich nun auf `storageMedium.p_default` statt auf den ungültigen Klassenzugriff `fluidDom.Medium.p_default`.

Kleines Beispiel (Alpha)

AI ASSISTANT

Verifiziere die Lösung durch eine Simulation von
LargeTESmtk.TankTES.Cylinder.Examples.FullyBuried_Basic.UniformGround

✓ Loading skill instructions 0.5s

✓ Checking model class type 2.2s

✓ Fetching experiment definitions 1.3s

Nur eine Experiment-Definition ("Experiment 1"), passend zum aktuellen Kontext — keine Überschreibungen angefragt, daher starte ich direkt.

✓ Running simulation 1.1s

✓ Waiting for simulation 7.8s

Die Simulation ist fehlgeschlagen. Ich prüfe das Log, um zu sehen, ob der ursprüngliche Fehler behoben ist oder ein neues Problem vorliegt.

✓ Fetching simulation log 0.5s

Der ursprüngliche Fehler ist behoben: Die Meldung „Cannot access non-function class through component access: fluidDom.Medium.p_default“ tritt in diesem Lauf **nicht mehr auf**. Die Simulation schlägt jedoch weiterhin fehl — an anderer, unabhängiger Stelle.

Status: The simulation failed.

Verbleibende Fehler (bereits im vorherigen Log als Folgefehler vermutet, bestehen aber unverändert weiter — sie sind also eigenständige, vom p_start-Fix unabhängige Probleme):

- Cannot find declaration for r in LargeTESmtk.Components.FluidDomain.Utilities.SegmentDim.PartialSegDim (betrifft storage.fluidDom.fluidDomDim, nodeDim[10..] u.a.)
- Mehrere Equality comparisons do not allow real operands-Fehler in LargeTESmtk.Components.GroundDomain.GridUnits... (z. B. r_le == 0, GR_r_act == 1, GR_z_act == 1)
- Ein Typfehler beim Aufruf von transpose() in GridZone_Rect (thermProp_spec hat falsche Dimension)

Kleines Beispiel (Alpha)

AI ASSISTANT

Das sind strukturelle Probleme in der GroundDomain/FluidDomain-Bibliothek, nicht Folgefehler der p_start-Korrektur.

Der ursprüngliche p_start-Fehler ist behoben, aber es bestehen weitere, unabhängige Kompilierfehler (u. a. 'Cannot find declaration for r' und mehrere 'Equality comparisons do not allow real operands'-Fehler in GroundDomain/FluidDomain). Soll ich diese Fehler jetzt analysieren?

Ja, analysieren

✓ Fetching model source 2 calls 1.0s

Ergebnis der Verifikation

Die vorherige Korrektur wirkt: Der ursprüngliche Fehler „Cannot access non-function class through component access: fluidDom.Medium.p_default“ tritt **nicht mehr auf**. Die Simulation schlägt jedoch weiterhin fehl — an mehreren, davon unabhängigen Stellen in der Bibliothek LargeTESmtk.Components.GroundDomain und FluidDomain.

Empfehlung

Diese verbleibenden Fehler sind **eigenständige Bibliotheksdefekte**, keine Folgefehler des p_start-Fixes. Sie erfordern Änderungen an mehreren Modellklassen und sollten dem Bibliotheksverantwortlichen gemeldet werden, bevor das Beispielmmodell erfolgreich kompiliert.

Fundierte Ursachen

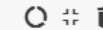
1. Ungültige Gleichheitsvergleiche auf Real-Werten (Hauptursache, betrifft die meisten Meldungen)

In PartialGridSegment_Rect, ZoneLayout_Radial, ZoneLayout_Axial und SegmentLayout_Radial/_Axial stehen Ausdrücke wie:

```
final parameter Real r_le_act = if r_le==0 then Modelica.Constants.eps else r_le;  
final parameter Real GR_r_act = if reversedDir_GR_r then 1/GR_r else GR_r;  
...  
if GR_r_act==1 then ...
```

Kleines Beispiel (Alpha)

AI ASSISTANT



Die Modelica-Sprachspezifikation erlaubt die Vergleichsoperatoren `==` und `<>` (Gleichheit/Ungleichheit) **nicht** für Werte vom Typ `Real`, da exakte Gleichheit bei Fließkommazahlen numerisch unzuverlässig ist. Der Compiler von Modelon Impact setzt diese Regel strikt durch und lehnt jeden Vergleich wie `r_le==0`, `GR_r_act==1`, `GR_z_act==1` und ... `<> 0` ab. Das erklärt die Fehler in `r_le`, `GR_r_act` und `GR_z_act`.

2. Fehlende Deklaration von `r` in `PartialSegDim`

`PartialSegDim` berechnet `A_r` und `A_z_to` unter Verwendung einer Variablen `r`, die aber erst in der abgeleiteten Record-Klasse `SegDim_Cylinder` deklariert wird. Trotz der Redeklaration von `fluidDomDim/nodeDim` auf `SegDim_Cylinder` in `Cylinder` löst der Compiler `r` bei mehreren Komponenten (`storage.fluidDom.nodeDim[10..]` u.a.) nicht korrekt auf — vermutlich ein Problem in der Redeclare-Verkettung über mehrere Vererbungsebenen (`PartialFluidDomain` → `Cylinder` → `PartialXTES_Cylinder_FullyBuried_CustomGD`).

3. Typinkompatibilität bei `transpose()`

In `GridZone_Rect` wird `transpose(thermProp_spec)` aufgerufen, wobei `thermProp_spec` vom Typ `PartialThermProp[1,1]` (ein Record-Array) statt einer numerischen Matrix ist — `transpose()` erwartet eine reine Zahlenmatrix.

Nächster Schritt

Diese drei Defekte liegen tief in der Bibliotheksstruktur (mehrere Klassen, mehrere Dateien) und sind keine einzelnen Parameterkorrekturen wie zuvor. Möchten Sie, dass ich einen konkreten Korrekturvorschlag für die Real-Gleichheitsvergleiche (Punkt 1) ausarbeite, oder sollen diese Punkte an den Bibliotheksverantwortlichen weitergegeben werden?

ausarbeiten

Kleines Beispiel (Alpha)

AI ASSISTANT

Korrekturvorschlag: Ungültige Real-Gleichheitsvergleiche

Die Modelica-Sprachspezifikation verbietet == und <> auf Werten vom Typ Real. Betroffen sind **6 Dateien** mit insgesamt 8 Vergleichsausdrücken. Ich schlage vor, jeden ==/<>-Vergleich durch einen toleranzbasierten Vergleich mit Modelica.Constants.eps zu ersetzen — das ist die in Modelica übliche Praxis für "ist praktisch gleich".

Datei	Aktueller Ausdruck	Vorgeschlagene Änderung
PartialGridSegment_Rect	if r_le==0 then Modelica.Constants.eps else r_le	if abs(r_le) < Modelica.Constants.eps then Modelica.Constants.eps else r_le
PartialGridSegment_Rect	if GR_r_act==1 then	if abs(GR_r_act-1) < Modelica.Constants.eps then
PartialGridSegment_Rect	if GR_z_act==1 then	if abs(GR_z_act-1) < Modelica.Constants.eps then
PartialNodeInterior	if r_le==0 then Modelica.Constants.eps else r_le	if abs(r_le) < Modelica.Constants.eps then Modelica.Constants.eps else r_le
SegmentLayoutRadial	if r_le==0 then Modelica.Constants.eps else r_le	if abs(r_le) < Modelica.Constants.eps then Modelica.Constants.eps else r_le
SegmentLayoutRadial	if GR_r_act==1 then	if abs(GR_r_act-1) < Modelica.Constants.eps then



Kleines Beispiel (Alpha)

AI ASSISTANT

SegmentLayout t_Axial	if GR_z_act==1 then	if abs(GR_z_act-1) < Modelica.Constants.eps then
ZoneLayout_R adial	if r_le==0 then Modelica.Constants.eps else r_le	if abs(r_le) < Modelica.Constants.eps then Modelica.Constants.eps else r_le
ZoneLayout_A xial	if ...setValue <> 0 then (...) else 0 (x2, für z_Sec_to_glo und z_Sec_bo_glo)	if abs(...setValue) > Modelica.Constants.eps then (...) else 0

Erklärung: Ein Vergleich wie `r_le==0` prüft auf exakte Bitgleichheit mit Null, was bei berechneten Fließkommazahlen unzuverlässig ist und vom Modelica-Compiler grundsätzlich abgelehnt wird. `abs(r_le) < Modelica.Constants.eps` prüft stattdessen, ob der Wert innerhalb einer verschwindend kleinen Toleranz bei Null liegt — semantisch gleichwertig zur ursprünglichen Absicht, aber sprachkonform.

Das behebt ausschließlich die Fehlerklasse "Equality comparisons do not allow real operands" (8 der 10 Fehlermeldungen). Die verbleibenden zwei Probleme — "Cannot find declaration for r" und der `transpose()`-Typfehler — sind strukturell anders gelagert und würden separat behoben.

Soll ich diese 9 Codeänderungen in den 6 Dateien jetzt vornehmen?

ja

✓ Running script 3.1s

✓ Running script 14.4s

✓ Running script 2.7s

Alle 9 Änderungen wurden in den 6 Dateien vorgenommen — jeder ungültige `==/<>`-Vergleich auf einem Real-Wert ist durch einen toleranzbasierten `abs(...) < Modelica.Constants.eps` bzw. `abs(...) > Modelica.Constants.eps`-Vergleich ersetzt:

Kleines Beispiel (Alpha)

AI ASSISTANT

- PartialGridSegment_Rect.mo — 3 Stellen (r_le_act, GR_r_act, GR_z_act)
- PartialNode_Interior.mo — 1 Stelle (r_le_act)
- SegmentLayout_Radial.mo — 2 Stellen (r_le_act, GR_r_act)
- SegmentLayout_Axial.mo — 1 Stelle (GR_z_act)
- ZoneLayout_Radial.mo — 1 Stelle (r_le_act)
- ZoneLayout_Axial.mo — 2 Stellen (z_Sec_to_glo, z_Sec_bo_glo)

Möchten Sie, dass ich die Simulation von LargeTESmtk.TankTES.Cylinder.Examples.FullyBuried_Basic.UniformGround erneut ausführe, um zu prüfen, ob diese Fehlerklasse behoben ist?

ja

✓ Checking model class type 0.8s

✓ Fetching experiment definitions 0.7s

Nur eine Experiment-Definition, passend zum aktuellen Kontext, keine Überschreibungen angefragt — ich starte direkt.

✓ Running simulation 1.0s

✓ Waiting for simulation 11.4s

The simulation failed. Ich prüfe das Log, um zu sehen, welche Fehler noch bestehen.

✓ Fetching simulation log 0.6s

Verifikationsergebnis

Der Fix wirkt: Alle 8 „Equality comparisons do not allow real operands“-Fehler sind verschwunden. Die Kompilierung schlägt jetzt nur noch an den zwei bereits zuvor identifizierten, **strukturell anderen** Stellen fehl:

Kleines Beispiel (Alpha)

AI ASSISTANT

1. **Cannot find declaration for r** in `LargeTESmtk.Components.FluidDomain.Utilities.SegmentDim.PartialSegDim` (betrifft `storage.fluidDom.fluidDomDim,nodeDim[10..]` u. a. — 51 Komponenten)
2. **transpose()-Typfehler** in `LargeTESmtk.Components.GroundDomain.GridUnits.GridZone_Rect`: `thermProp_spec` ist ein Record-Array `PartialThermProp[1,1]`, aber `transpose()` erwartet eine numerische Matrix.

Beide waren bereits vor dem letzten Fix vorhanden und sind unabhängig von den Real-Gleichheitsvergleichen — sie erfordern eigene, gesonderte Korrekturen (unterschiedliche Klassen, andere Fehlerart).

Soll ich für diese beiden verbleibenden Fehler ebenfalls einen Korrekturvorschlag ausarbeiten?

ja

✓ Fetching model source 4 calls 0.6s

Korrekturvorschlag für die zwei verbleibenden Fehler

1. Cannot find declaration for r in PartialSegDim

Ursache: Die Basis-Record `PartialSegDim` definiert die Standardwerte von `A_r` und `A_z_to` unter Verwendung der Variable `r`:

```
parameter Modelica.Units.SI.Area A_r = 2*r*Modelica.Constants.pi*dz "...";
parameter Modelica.Units.SI.Area A_z_to = r^2*Modelica.Constants.pi "...";
```

`r` wird jedoch erst in der abgeleiteten Klasse `SegDim_Cylinder` deklariert (parameter `Modelica.Units.SI.Radius r`). Eine Basisklasse darf in Modelica keinen Bezeichner verwenden, der erst in einer abgeleiteten Klasse existiert — das verletzt die Regel, dass jede Klasse für sich flach auflösbar sein muss. `SegDim_Cylinder` überschreibt diese Werte ohnehin bereits vollständig mit `final A_r=...`, `final A_z_to=...`, sodass die `r`-abhängigen Standardwerte in `PartialSegDim` nie tatsächlich verwendet werden — sie sind toter Code, der aber trotzdem geprüft wird und den Fehler auslöst.

Kleines Beispiel (Alpha)

AI ASSISTANT

Vorgeschlagene Änderung in PartialSegDim: die Standardwerte entfernen, die auf `r` verweisen, und die Parameter ohne Default deklarieren (jede konkrete Geometrie-Unterklasse liefert den Wert über `final`-Modifizier, wie `SegDim_Cylinder` es bereits tut):

```
parameter Modelica.Units.SI.Area A_r "Lateral boundary surface";
parameter Modelica.Units.SI.Area A_z_to "Top boundary surface perpendicular to axial direction";
``//(statt der beiden Zeilen mit `= 2*r*...` bzw. `= r^2*...`)
```

Die restlichen Zeilen (`[A_z_bo](model-parameter://A_z_bo)`, `[A](model-parameter://A)`, `[V](model-parameter:`

2. ``transpose()``-Typfehler in ``GridZone_Rect``

****Ursache:**** ``thermProp_spec`` ist ein Array von Records (``PartialThermProp[N_Sec_z, M_Sec_r]``), kein nume

****Vorgeschlagene Änderung:**** den ``transpose()``-Aufruf durch einen expliziten Array-Konstruktor ersetzen,

```
final parameter LargeTESmtk.Components.GroundDomain.MaterialProperties.BaseClasses.PartialThermProp
thermProp[zoneLayoutPar_r.M_Sec_r, zoneLayoutPar_z.N_Sec_z]=if useUnifThermProp then thermProp_unif_matrix else
array(thermProp_spec[n, m] for m in 1:zoneLayoutPar_r.M_Sec_r, n in 1:zoneLayoutPar_z.N_Sec_z) "Thermophysical properties of
individual segments in grid zone" annotation (Dialog(group="Ground properties"));

`array(expr for m in ..., n in ...)` erzeugt hier ein `[M_Sec_r, N_Sec_z]`-Array mit `thermProp[m,n] = th
---
```

Soll ich beide Änderungen jetzt im Quellcode vornehmen?

ja

Kleines Beispiel (Alpha)

AI ASSISTANT

Soll ich beide Änderungen jetzt im Quellcode vornehmen?

ja

✓ Running script 2.9s

✓ Running script 6.1s

✓ Running script 1.0s

Beide Änderungen sind angewendet:

- **PartialSegDim.mo** — `A_r` und `A_z_to` sind jetzt ohne Default-Wert deklariert, statt auf das erst in der Unterklasse existierende `r` zu verweisen.
- **GridZone_Rect.mo** — der `transpose(thermProp_spec)`-Aufruf ist durch einen expliziten `array(...)`-Konstruktor ersetzt, der Record-Arrays korrekt transponiert.

Möchten Sie, dass ich die Simulation erneut ausführe, um zu verifizieren, ob das Modell jetzt kompiliert?

ja

✓ Running simulation 1.6s

✓ Waiting for simulation 176.9s

Simulation erfolgreich abgeschlossen. Alle drei Fehlerklassen sind behoben:

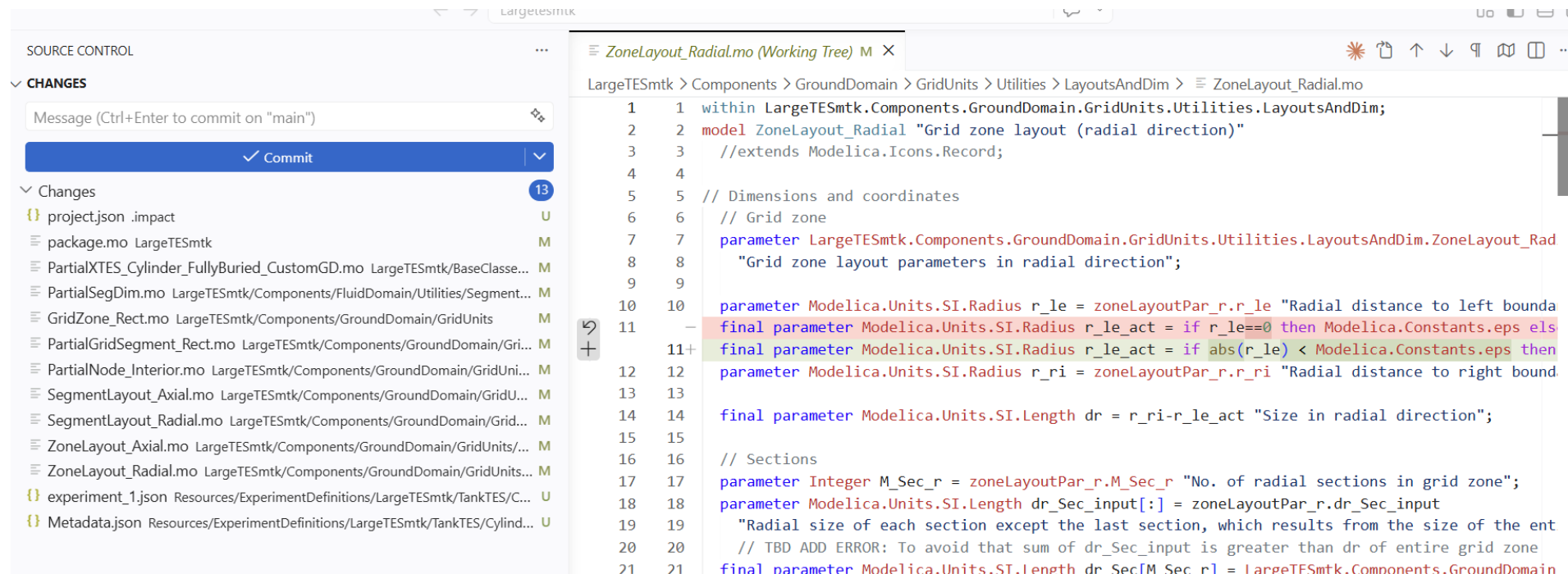
1. Ungültiger Klassenzugriff `fluidDom.Medium.p_default` → korrigiert auf `storageMedium.p_default`
2. Ungültige Real-Gleichheitsvergleiche (`==/<>`) → toleranzbasiert mit `Modelica.Constants.eps` ersetzt
3. Fehlende `r`-Deklaration in `PartialSegDim` und `transpose()`-Typfehler in `GridZone_Rect` → behoben

LargeTESmtk.TankTES.Cylinder.Examples.FullyBuried_Basic.UniformGround kompiliert und simuliert jetzt fehlerfrei. Sie können die Ergebnisse in der Modelon Impact UI ansehen, oder mich bitten, eine bestimmte Ergebnisvariable zu analysieren.

Kleines Beispiel (Alpha)

- 18 Minuten (inkl. Slide-Updates und Zähneputzen)

SIMULATIONS		
	Result 4	00:14
	Result 3 (Failed compilation)	00:11
	Result 2 (Failed compilation)	00:03
	Result 1 (Failed compilation)	Sep 22 23:56



The screenshot displays the ThermoSim 2026 IDE interface. On the left, the 'SOURCE CONTROL' panel shows a list of changes under the 'CHANGES' tab, including files like 'project.json', 'package.mo', and 'ZoneLayout_Radial.mo'. A 'Commit' button is visible. The main editor window shows the 'ZoneLayout_Radial.mo' model in the 'Working Tree'. The code defines a model for a radial grid zone layout, including parameters for dimensions, coordinates, and radial distances, and final parameters for the grid zone layout.

```
1 1 within LargeTESmtk.Components.GroundDomain.GridUnits.Utilities.LayoutsAndDim;  
2 2 model ZoneLayout_Radial "Grid zone layout (radial direction)"  
3 3 //extends Modelica.Icons.Record;  
4 4  
5 5 // Dimensions and coordinates  
6 6 // Grid zone  
7 7 parameter LargeTESmtk.Components.GroundDomain.GridUnits.Utilities.LayoutsAndDim.ZoneLayout_Rad  
8 8 "Grid zone layout parameters in radial direction";  
9 9  
10 10 parameter Modelica.Units.SI.Radius r_le = zoneLayoutPar_r.r_le "Radial distance to left bounda  
11 11 final parameter Modelica.Units.SI.Radius r_le_act = if r_le==0 then Modelica.Constants.eps els  
11+ 11 final parameter Modelica.Units.SI.Radius r_le_act = if abs(r_le) < Modelica.Constants.eps then  
12 12 parameter Modelica.Units.SI.Radius r_ri = zoneLayoutPar_r.r_ri "Radial distance to right bound  
13 13  
14 14 final parameter Modelica.Units.SI.Length dr = r_ri-r_le_act "Size in radial direction";  
15 15  
16 16 // Sections  
17 17 parameter Integer M_Sec_r = zoneLayoutPar_r.M_Sec_r "No. of radial sections in grid zone";  
18 18 parameter Modelica.Units.SI.Length dr_Sec_input[:] = zoneLayoutPar_r.dr_Sec_input  
19 19 "Radial size of each section except the last section, which results from the size of the ent  
20 20 // TBD ADD ERROR: To avoid that sum of dr_Sec_input is greater than dr of entire grid zone  
21 21 final parameter Modelica.Units.SI.Length dr_Sec[M_Sec_r] = LargeTESmtk.Components.GroundDomain
```

Generative KI sinnvoll nutzen – wie?

Tools

- Sicherheit!
- Nachvollziehbarkeit
- Verifizierung
- Sanity-Checks / „Mitdenken“

⚠ **Wichtiger Hinweis zur Machbarkeit:** Eine Jahressimulation mit zwei detaillierten, zweiphasigen Kältemittelkreisläufen (VaporCycle + ThermofluidStream) parallel zu einer Carnot-Wärmepumpe ist numerisch sehr anspruchsvoll — 8760 h mit wechselnden Randbedingungen können lange Rechenzeiten und Konvergenzprobleme verursachen. Ich empfehle, das Modell zunächst an einer kurzen Testperiode (z. B. eine Winter- und eine Sommerwoche) zu validieren, bevor die volle Jahressimulation gestartet wird.



Der VaporCycle-Zweig kompiliert und simuliert erfolgreich — die 24-h-Simulation des kältesten Tages ist durchgelaufen.

Kurzer Sanity-Check der Ergebnisse, bevor ich zum ThermofluidStream-Zweig weitergehe:

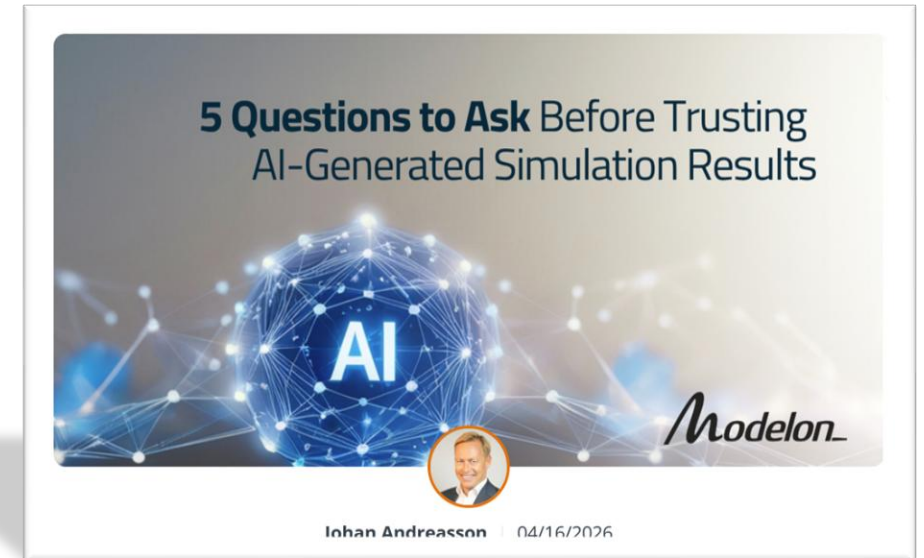
✓ Fetching result trajectories 1.9s

Die Simulation lief nur bis **t=800 s** statt der vorgesehenen 24 h (86400 s) — die genutzte Experimentdefinition "Fluid" speichert offenbar noch die alte Stoppzeit aus der ursprünglichen Vorlage. Ich prüfe und korrigiere das:

Generative KI sinnvoll nutzen - wie?

Nutzer

- Flaschenhals nicht mehr Produktion, sondern Review → Qualität statt Quantität
- Versionskontrolle
- Richtig prompten
- Offenheit und Anpassungsfähigkeit



Die Reise geht weiter...

Wer kommt mit?

