



# Tikos Reasoning Platform: A Case Study



Mike Oaten

Published in Tikos Tech

7 min read · 4 days ago



Listen



Share

⋮ More

Tikos takes on a cancer analysis.

## Introduction

*Tikos is developing AI reasoning technology which delivers:*

- i) Advance reasoning,*
- ii) Explainable outputs, and*
- iii) High-performance processing*

These capabilities are built on a family of formal methodologies, mathematical techniques and algorithms (patents sought) that specifically implement a case-based (abductive) reasoning system.

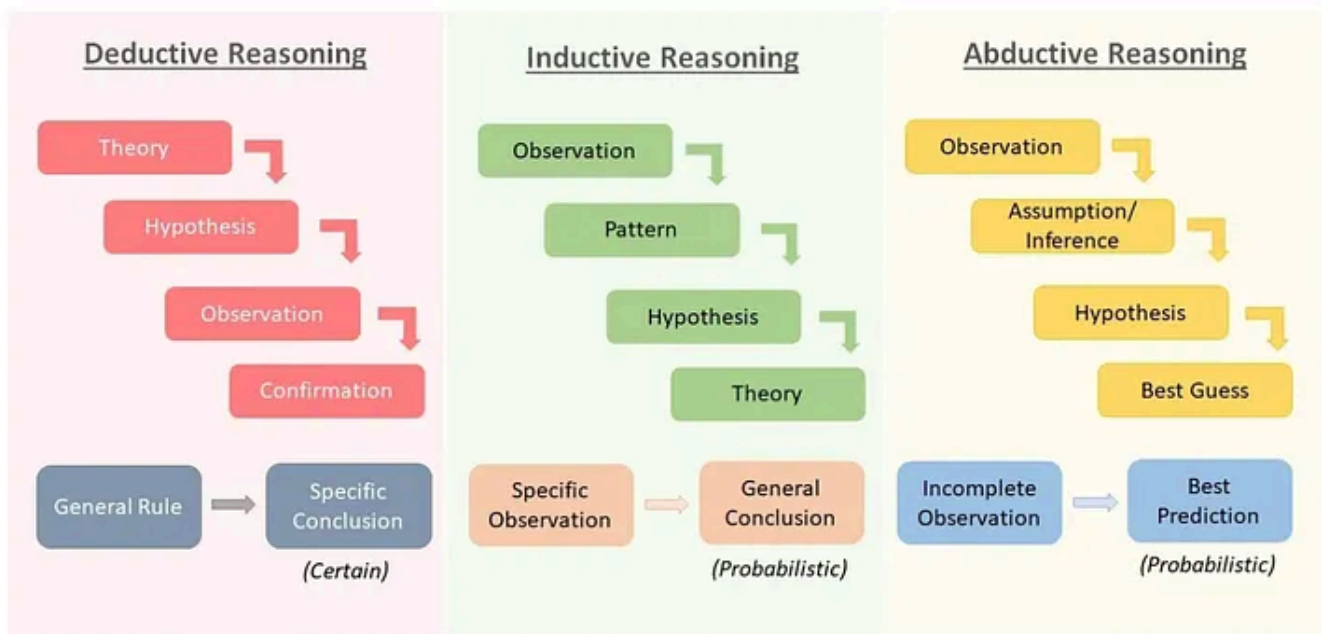
Before showing the Tikos solution in action, let's explain case-based reasoning.

### **What is case-based (abductive) reasoning?**

In case-based reasoning systems, the primary knowledge source is a memory of preserved cases, which records specific information from prior experiences. In CBR systems, reasoning is based on searching and retrieving appropriate cases and adapting them to solve the problem at hand. Importantly, cases are not decomposed into their constituent features but dealt with as a whole — a case.

Imagine standing in front of a door you've never seen before and needing to open it. You call to mind similar experiences and soon enough try the handle and find it locked, look under a nearby plant pot, locate a spare key, unlock the door and pass through. Congratulations! You have successfully employed case-based reasoning.

Case-based (abductive) reasoning is one of the big three approaches...



<https://design-thinking.in/types-of-reasoning-1>

**Deductive approaches** use logic, where facts and proven rules are combined to progress from general theory to specific conclusions. While outputs can be explained, deductive approaches do not adapt easily to novel problems and are limited to tasks that can be codified in formal logic (e.g. Prolog, Datalog).

**Inductive approaches** use historical data, patterns, or associations to progress from specific observations to general conclusions. Many machine learning methods and neural networks are examples. Whilst often statistically impressive, outputs are not reliable or accurate in the individual case and for complex models, rarely explainable.

**Case-based reasoning (abductive)** is suitable for real-world complex problem-solving because it is a creative process that can adapt existing known cases to novel problems, remember its performance and modify its behaviour. Additionally, the processes and output are always transparent and explainable.

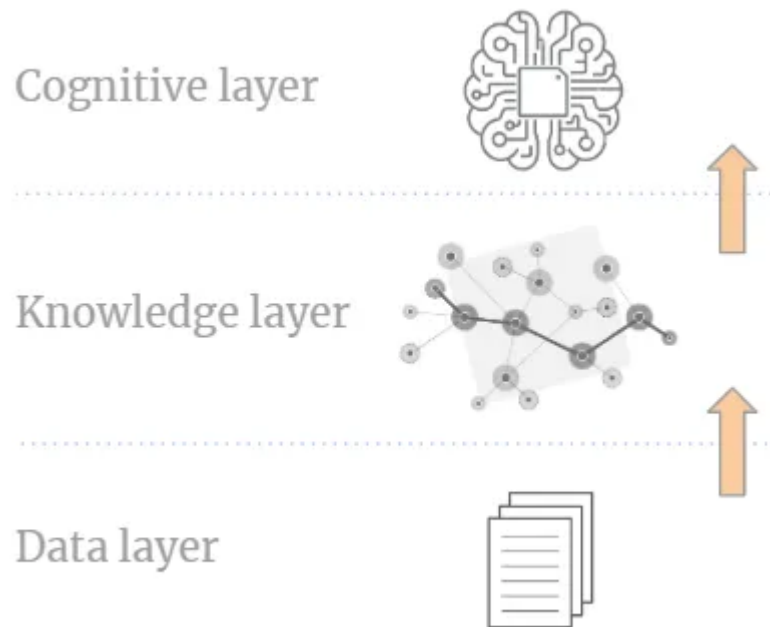
Commercial (machine) implementations of CBR have not kept pace with advances in AI and computing, unlike rule-based and inductive approaches which have. This is due to the problem being — or perceived to being — too challenging.

Now we've set the scene, let us move on to setting up Tikos.

**Tikos requirement #1 — Context**

Firstly, Tikos needs a contextual model of the domain. In practice, this simply means adding or connecting relevant source data or documents to the system.

Tikos converts the source data (the data layer) into a symbolic representation (the knowledge layer) over which ‘context-aware’ reasoning functions operate (the cognitive layer).



Tikos reasoning function operate at the cognitive layer

### **Tikos requirement #2 — Cases**

Second, cases need to be defined. Cases are simply a standard set of data values. This is a simple design choice undertaken by most stakeholders, it is not restricted to technical staff or subject matter experts. Tikos performs well where a ‘case’ is the intuitive way to frame knowledge for a problem/solution situation, for example:

*Personalised marketing: Historical purchase patterns are used to retrieve similar cases based on customer preferences, behaviors, and demographics.*

*Financial risk assessment: Past cases of loan applicants with similar credit histories, incomes, and employment statuses.*

*Fault diagnosis: Past maintenance and fault logs are used to identify cases similar to the current machinery issue.*

*Customer support: Past customer inquiries and resolutions are retrieved to offer a potential solution for a current issue.*

*Pricing and demand forecasting: Historical cases of product pricing and demand under similar market conditions are retrieved.*

*Legal cases for contract reviews: The system matches new contract cases with previously resolved legal disputes or contract reviews.*

## **The Tikos Reasoning Platform in action**

The following is a worked example of a cancer analysis. Hospital departments will already have the raw ingredients to implement the Tikos solution — lots of relevant source data.

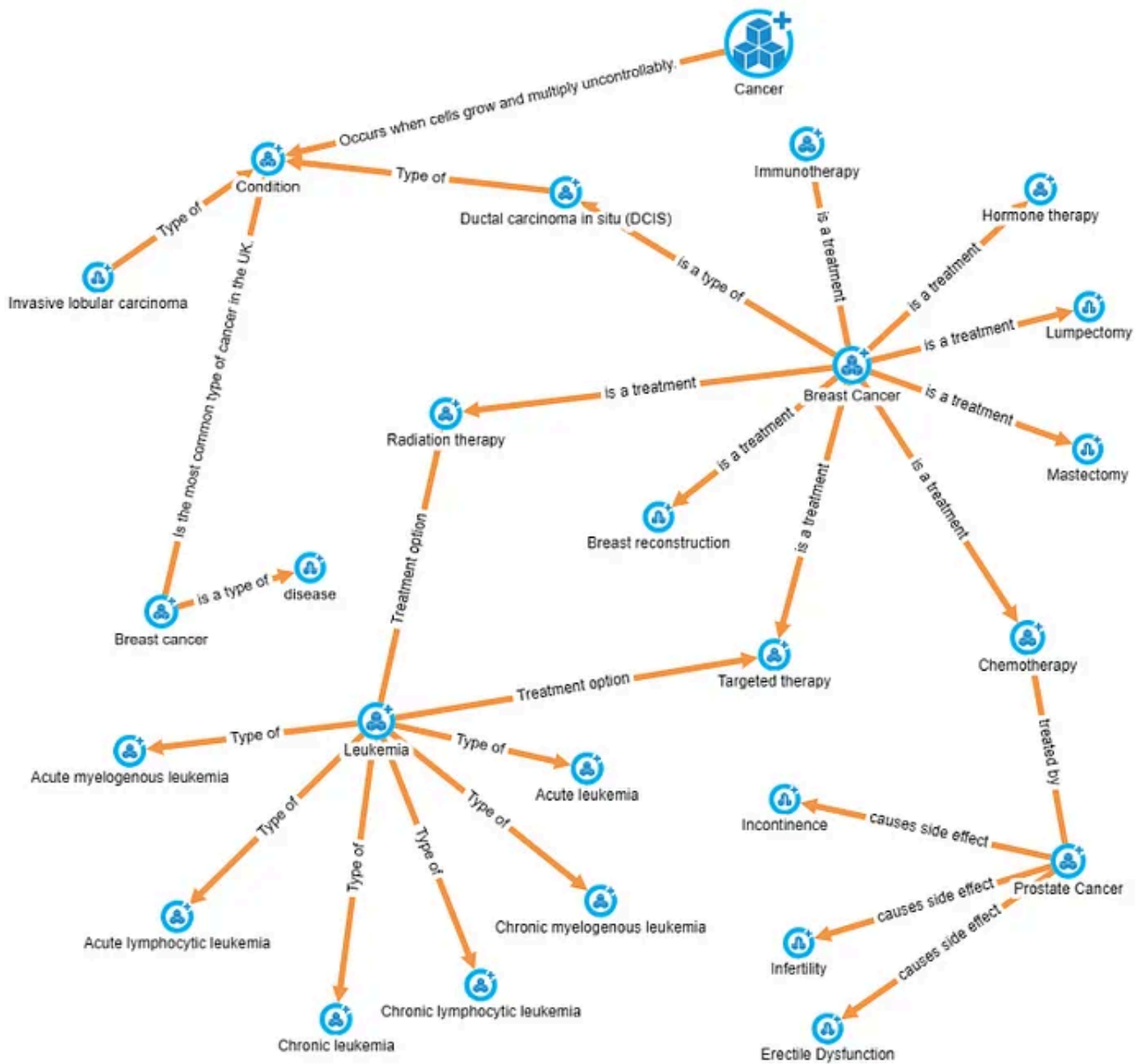
### **Step 1 —Define the problem**

“Understand the primary cause of disease for a patient’s cancer diagnosis.”

### **Step 2—Create the context**

The next step is to create a contextual model for the reasoning engine. While the Tikos solution is datastore and data-model agnostic, capturing a symbolic representation of the relevant knowledge in a graph structure provides a rich context for the reasoning operations to operate. In some cases, relevant knowledge graphs may exist or otherwise can be created using Tikos.

The image below shows a small fraction of a knowledge graph for our cancer analysis. In production, the knowledge graph could run to hundreds of thousands of entities, relationships and properties.



A very small section of the context — a knowledge graph

### Step 3— Collate relevant cases

In this example, we're using 500 cancer cases (synthetic data) compiled in a consistent key:value format, a pair for each of the 25 case items ('case-item' — not 'feature' — being the preferred terminology). We want to understand the cause of disease for the following problem case:

```
"patient_demographics.age": 54,  
"patient_demographics.gender": "Female",  
"patient_demographics.smoking_history": false,  
"patient_demographics.family_history": true,  
"diagnosis.cancer_type": "Breast",  
"diagnosis.stage": "Stage IV",  
"diagnosis.grade": 3,  
"diagnosis.tumor_size_cm": 4.2,  
"diagnosis.metastasis": true,  
"diagnosis.lymph_node_involvement": true,  
"clinical_presentation.symptoms": "Fever, Nausea, Pain, Weight Loss, Cough",  
"clinical_presentation.duration_months": 2,  
"clinical_presentation.performance_status": 1,  
"treatment.plan": "Surgery, Stem Cell Transplant, Targeted Therapy",  
"treatment.start_date": "2020-10-13",  
"treatment.end_date": "2021-10-02",  
"treatment.response_rate": 91,  
"treatment.side_effects": "",  
"outcome.status": "Partial Remission",  
"outcome.survival_months": 35,  
"outcome.quality_of_life_score": 3,  
"outcome.complication_rate": 21.6,  
"biomarkers.marker1_level": 76.25,  
"biomarkers.marker2_level": 67.96,  
"biomarkers.genetic_mutation": true
```

The problem case

The individual case data object can be created from data in any system and Tikos does not dictate its design, but generally the more relevant data the better. This is particularly important when the system is answering complex questions and where it is implemented with a neural network (the intended approach for most production deployments).

#### **Step 4— Create the Sequential Collection**

Sequential Collection is a proprietary data structure used to serialise each case and with reference to the whole case base (the 500 cancer cases). The elements in this data structure are optimised for indexing, case similarity matching, case adaption, reasoning and explainability functions which can be performed efficiently at scale.

T	1
N	0
w	1.0
3	0
Γ	["d0c148a7-f7c0-4a4c-9df4-db1946d3619c", "...
Γi	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ...
Γa	["d0c148a7-f7c0-4a4c-9df4-db1946d3619c", "...
S	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ...
W	["d0c148a7-f7c0-4a4c-9df4-db1946d3619c", "...
Wv	[16800039, 69, 6502514, 1, 1, 8751434, 75314...
t	["str", "int", "str", "bool", "bool", "str", "str", "int...
Vmin	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
Vmax	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
l	["case_id", "patient_demographics.age", "patie...
c	[]
P	[1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1...

*(side note) Tikos can be implemented in two modes:*

*Naively, with traditional data stores/analysis (small scale/low complexity cases).*

*Neurally, as part of a specialised neural network architecture which is the intended mode for most production systems.*

Here we show the naive implementation, where the input for the Sequential Collection are the case data items. In neural implementations, the input for the Sequential Collection are feature activation maps from neural network training.

*We now have all the elements for Tikos to perform its reasoning functions...*

### **Step 5— Case similarity search**

Tikos indexes the case base.

Then, using our proprietary similarity search algorithms, Tikos identifies the top n cases from the case base matching to the problem case. It is worth reiterating that the similarity functions do not decompose cases into their constituent case items but treat them as single entities.

The returned set of one or more cases from the solution case base represents those closest prior experiences to the problem case we are working on.

## Step 6— Case adaptation

The returned set of candidate solution cases are then compared to the problem case and each other. This is to understand the causal inter-relationships between case items and then make adaptations to create the final solution case. The aggregate adaptation map for our final solution case:

```
0.0030117511050538137,  
0.0023925886482385654,  
0.0,  
0.007071067811865476,  
0.004758369728464714,  
8.552202428658365e-06,  
0.0023570226039551587,  
0.0057241977524625275,  
0.007071067811865476,  
0.007071067811865476,  
0.0,  
0.0,  
0.0,  
0.0,  
0.0,  
0.0,  
0.0,  
0.0,  
0.0,  
0.00020203050891044218,  
0.0,  
0.0,  
0.004758369728464714,  
0.0,  
0.0
```

Aggregate adaption map

## Step 7— Final solution case

This is a case that best answers the original problem. It is returned along with the ‘case item causation probability’ which determines how much the specific case item is causally relevant in the adaptation process.

The case is then added to the case base (tagged as an adapted case) along with all its associated data and processing from the search and adaptation steps. This way, Tikos ‘learns’ and becomes more useful with every user interaction.

## Step 8— Reasoning and explainability

We now call on the knowledge graph created earlier (our context) and an open-source large language model.



We want to understand why the previous variations and adaptations were suggested for the final solution case. We achieve this by using each case item's causation probabilities and adaptation weights to guide a retrieval augmentation generation across the knowledge graph (graphRAG).

Below is the output for case item #23 'biomarkers.marker1\_level':

```
" The relationship between biomarkers, specifically Marker2 (HER2), in breast cancer is that overexpression (HER2-positive) requires specific targeted therapies like trastuzumab. Having equal relationships of Marker2 level could imply similar response to HER2-targeted treatments. However, this information is based solely on the provided sources and may not apply to other cancer types or biomarkers.",
```

In total, we have 25 'weighted' explanations, one for each case item. Finally, the explanations are summarised by the language model:

```
"The cancer case involves Stage IV breast cancer in a 54-year-old female with a family history of cancer, indicating a genetic predisposition. Biomarkers play a critical role, particularly HER2 overexpression (Marker2), which necessitates targeted therapy like trastuzumab. The high Marker1 level also aligns with hormone receptor positivity, potentially influencing prognosis and treatment strategies."
```

## Summary

We applied the Tikos Reasoning Platform to a cancer case analysis and demonstrated:

- **Context-aware reasoning:** Tikos implemented a contextual model (a knowledge graph) to enable reasoning for a complex problem grounded in the organisation's data.
- **Explainable AI:** Tikos provided transparent explanations, enabling users to understand and trust the reasoning and adaptation processes leading to the final output and for all intermediate steps.
- **Flexibility and versatility:** Tikos gave users freedom to design the case data object, and select the case base and source data for the contextual model. Users also had an option to use either a traditional or neural network implementation.
- **Self-learning capability:** Tikos added the adapted solution case and associated data back into the case base, improving its reasoning, solution accuracy and

value over time with every user interaction.

## AI Agents

The Tikos AI reasoning and explainability platform is perfect for developers creating agentic systems that must earn the trust of users, system owners, and regulators. Development teams require tools that can demonstrate they are safe, smart and compliant from the start. Tikos is designed around these very principles.

## What Next?

Check out our Python library for running the end-to-end pipeline  
<https://tikos.readme.io/>

Join the waitlist by emailing  
[devs@tikos.tech](mailto:devs@tikos.tech)

For all product and company updates follow Tikos  
<https://www.linkedin.com/company/tikos-tech/>

Or email me  
[mike.oaten@tikos.tech](mailto:mike.oaten@tikos.tech)

Aireasoning

Explainable Ai

Knowledge Graph

Llm

Case Based Reasoning



Following

Published in Tikos Tech