Agentic systems
Lessons learned

Jan Křivánek

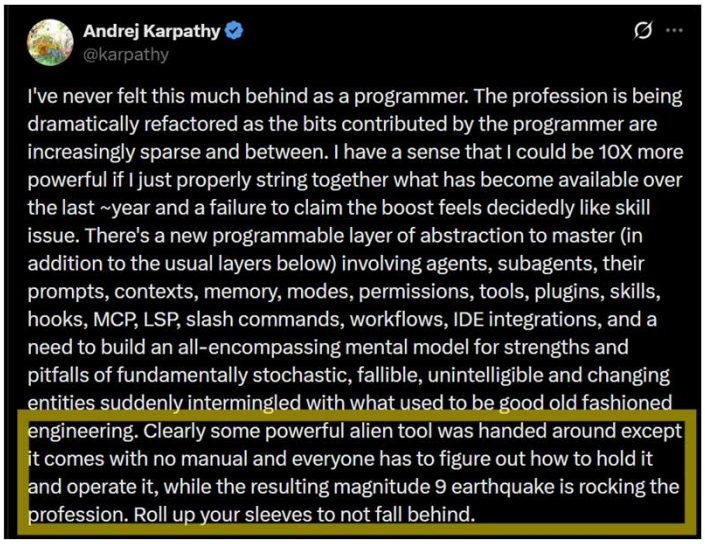
Content

- Intro
- Context engineering
  - Motivation
  - Techniques
- Tools engineering
  - Improving
  - Special tools
- Agentic workflows
- Vibe coding

Intro

dotutils.net/wug-talk

FOMO! Question, Revalidate



https://x.com/karpathy/status/2004607146781278521

Study sources

- https://www.youtube.com/@AndrejKarpathy/videos
- Anthropic
  - https://www.anthropic.com/engineering
  - https://www.anthropic.com/research
- https://steipete.me/posts/2025/shipping-at-inference-speed
- …

What is agentic AI?

- Various definitions
  - Limited supervision needed for tasks completions
  - Full autonomy vs fixed workflow
  - Single- vs multi-agent systems
  - Anthropic: **LLM autonomously using tools in a loop**
- For our purpose
  - **LLM is called with input, created based on another LLM call**
- Covers
  - Multi-step processing
  - Tools calling
  - Planning / Orchestrating
  - A2A

https://www.anthropic.com/engineering/building-effective-agents

Quiz - # of times system prompt was evaluated

- System:
  - You are helpful assistant …
  - Use **read_file**(name, startLine, endLine) tool to read at **max 10 lines** of file at a time
- User:
  - Add numbers on first **25 lines** in numbers.txt

…

?

a) 1x
b) 2x
c) 3x
d) 4x
e) ???

Context engineering

Context engineering - motivation

- API Hard Failures
- Output truncation
- Context Rot
- Context Anxiety

What is context and why we need it

https://research.trychroma.com/context-rot
+ needle in haystack
- Attention issues – The Transform NN - relations between tokens in context – the further distances has weaker relations

Context engineering – best practices

- System prompts
  - Clear, concise
  - Variables towards the end (prompt/tokens caching)
  - Sectionize (xml/md)
- Tools
  - Understandable, brief
  - Only needed ones
    - SWE-Mini (bash), anthropic tools calling agent (bash, str_replace_editor)
    - Dynamic tools selection (virtual tools), Skills (skills.sh/)
- RAG
  - Context Retrieval
  - Prefer tools

https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents
Context retrieaval – chunks reranking

Context engineering – strategies

- Compaction (summarization)
  - M.E.AI IChatReducer
  - Custom - LogViewer sample
  - Target older tool calls
- Optimize tools for context (next section)
- Memory
  - .md files
  - Memory tools (can be just KV store backed by files)
- Problem partitioning
  - E.g. the Research-Plan-Execute

Impl: e.g. **MessageCountingChatReducer - https://learn.microsoft.com/en-us/dotnet/api/microsoft.extensions.ai.messagecountingchatreducer?view=net-10.0-pp**
Usage – ReducingChatClient - https://learn.microsoft.com/en-us/dotnet/api/microsoft.extensions.ai.reducingchatclient?view=net-10.0-pp

Tools

Tools Best Practices

- Concise, descriptive
- Versatile vs specialized (see CC or (Mini-)SWE system prompt)
- Tools selections if we have many
  - RAG
  - Semantic search
  - Virtual grouping tools
  - Skills

SWE bench – https://github.com/SWE-agent/SWE-agent/tree/main/tools

Antropic – best practices: https://www.anthropic.com/engineering/writing-tools-for-agents

Antropic tools calling agent (bash + editor_str_replace)
https://www.anthropic.com/engineering/swe-bench-sonnet

Tools Best Practices (contd.)

- Clear naming
- Prompt engineered spec/descriptions
- Context mindful responses (with instructions)
- Usable context (or instructions) in response
- Descriptive errors
- Evaluate, Improve (rinse & repeat)
- Do NOT just expose your APIs

SWE bench – no tools: https://github.com/SWE-bench/SWE-bench/blob/main/swebench/inference/make_datasets/create_instance.py

Antropic – best practices: https://www.anthropic.com/engineering/writing-tools-for-agents

Tools eval cookbook: https://platform.claude.com/cookbook/tool-evaluation-tool-evaluation

Carefull about IDs – GUIDs/UUIDS are not good idea – use meaningful or numeric

Tools – M.E.AI wrapping example

```
public class MyChatClient : DelegatingChatClient
{
    0 references | 0 changes | 0 authors, 0 changes
    public override async Task<ChatResponse> GetResponseAsync(IEnumerable<ChatMessage> messages, ChatOptions? options = nul
    {
        options ??= new ChatOptions();
        options.Tools = options?.Tools?.Select(t => (t is AIFunction func) ? new MyAiFunction(func) : t).ToList();
        return await InnerClient.GetResponseAsync(messages, options, cancellationToken);
    }


internal sealed class MyAiFunction(AIFunction innerFunction) : AIFunction
{
    0 references | 0 changes | 0 authors, 0 changes
    protected override async ValueTask<object?> InvokeCoreAsync(AIFunctionArguments arguments, Cancellatic
    {
        var correctedArguments = RemapArgumentsIfNeeded(arguments);
        object? result = await innerFunction.InvokeAsync(correctedArguments, cancellationToken).ConfigureA
        string? resultString = result as string ?? JsonSerializer.Serialize(result);
        // ... Truncate, Catalogize ...
        return resultString;
    }
```

Example – viewer:
https://github.com/JanKrivanek/MSBuildStructuredLog/blob/dev/jankrivanek/chat/src/StructuredLogger.LLM/Tools/MonitoredAIFunction.cs#L19

Tools – some special tools

- Think vs Reasoning
  - Anthropic recommends using extended thinking
- Memory tool
- Plan + progress tool
- Ask User tool

Agentic workflows

Data and control transitions

- Code -> LLM -> Code -> LLM …
- Options
  - LLM wrapped in tool
  - Getting the data and/or status via tool vs final output
- Dilemma of structured vs unstructured data

Vibe coding

Boosters

- Feeback loop can be game changer
  - Expose logic via CLI
  - Point to CLI debugger(s)
  - MCPs for GUI – Playwright, Puppeteer, …
- Maintain docs/specs
- Features backlog

Unsorted

Quiz answer

- Depends … ☺
  - LLMs are stateless – so each tool call response resends the whole context
  - But there are optimizations
    - Parallel tool calls (LLM can call multiple tools in single response)
    - Prompt caching (matching prompt prefix gets cache hit from previous inferences)
- So
  - 4 without prompt caching hits, and with no parallel tool calls
  - More – as above and if there were errors returned for some tool calls
    - Or it decided to fetch in smaller chunks
  - 2 with prompt caching hits
  - 2-3 without caching hit and with some tool calls grouped
  - 2-3 without optimizations, if it tried to fetch more and tool returned

- Claude System Prompts
- *"The conversation has unlimited context through automatic summarization“*
- SWE Agent tool
- Binlog Viewer agentic chat