HS 525 : Learning Sciences

Created	@July 26, 2025 9:59 PM
Contact	pranav.joshi@iitgn.ac.in
Creator	Pranav Joshi

IHLS 1,2

Overview

- learning science is a practical (use case facing) science, i.e. in the “Pasteur’s quadrant”
- It is based on anthropology, cognitive psychology, cognitive sciences, educational psychology and sociology
- Unlike cognitive science, we don’t think about parts of brain and their function. The brain is not the system, only part of it.
- The full system is made up of the individuals, the environment, the ways of interaction between the individuals, etc.
- Learning science uses dialectical reductionism. We break a complex action (say doing a dance step) into its constituent actions. Then we study the smaller actions in the context of it being used for the larger action and explain from the information gained how the bigger action is performed. The “dialect” is most likely the context (the bigger action) in which the smaller actions are studied.
- Most of our actions are “tool mediated”
- Similar to the cognitive revolution (which gave birth to cognitive science and psychology) or the quantum revolution (giving rise to QM), the learning sciences emerged because of inconsistency between theory and practice.

  But unlike them, the “practice” are the seemingly inconsistent results and explanations coming from many different disciplines, not the “ground truth” (like you usually have in an experimental science).
- We’ll study why people go to the different sources of information (internet, peers, books, formal degree, etc.)
- We try to design environments for the learners where they cannot use rote/memorised solutions/facts and have to go *just* out of their comfort zone. This is called the “Zone of Proximal Development” (ZPD).
- For a learner to be successful in a ZPD , he needs support(s) called *scaffold(s)*. A scaffold should lead to knowledge and dispositional outcomes that can be used even in a different setting than the learning environment.
- A scaffold is usually a “more knowledgable other” . This can be a book, a person, or chatGPT
- We also study the effects of timing and frequency of feedback and guidance provided by the scaffold for *inquiry learning* .
- Then we study the effect of introducing technology in informal learning eco-systems

- Games as learning environments
- “Maker movement” has 2 core features : authenticity and purpose in making
- There is a thing called “Computer Supported Intentional Learning Environment” which fosters knowledge-transformation rather than knowledge telling.
- Design Based Research (DBR) is a technique used a lot in learning sciences. The researcher designs the system, sees how it is doing (with statistics usually) and re-designs. This cycle goes on indefinitely. A bit like user acceptance testing.
- Finally, there is the actual statistics knowledge needed by the designer

▼ Aug 7

- Start by studying what experts are doing
- Kids are definitely not experts but neither are they defficient so they can still do simplified versions of the stuff that experts do.
- Timeline for changes in learning sciences:
  - Behaviourilsm (think of mind as black box and study in lab) and Pragmantism (study the effect of different policies in a classroom (system of minds) and design better policies) . This is like a miniature version of design science based learning.
  - Cognitive psychology (unbox the brain)
  - Learning as a design science (game supported learning and game design) .
  - New technologies (AI and stuff)
- You have to be empirical in learning sciences (datasets..yay!)
- Pasteur’s quadrant

Application\ Knowledge	Low	High
High	Edison’s quadrant (light bulb)	Pasteur’s quadrant (pasteurisation)
Low	Doesn’t exist	Bohr’s quadrant (atomic model)

- Randomised Controlled Trials (RCT) are cases where when the sampling was done from the full population
- Educational interventions can’t be generalised, so RCTs don’t work well and are the final backup plan.

▼ Aug 12

- complex systems are those with lot of interconnected pieces. So many that modeling the interactions is almost impossible. Thus, there are very sudden changes of the response, with small changes in the input.
- We want assessment of the learning and the design of a course to go hand in hand. Basically, we want to able to say to learners “after reading this, you’ll be able to”
- Adaptive learning and selection of scaffolds.
- Mixed methods = quantitative and qualitative.
  For example, cog sci is just quantitative. But case studies and discourse analysis is qualitative.
- In the initial iterations of a learning science loop, analysis is usually qualitative, while the later iterations are quantitative.
- The researcher is not necessarily the more knowledgeable person.

▼ Cognition, Construction of Knowledge and Teaching

- Non-constructivists define knowledge as “justifiable true beliefs”
- Constructivism is a form of pragmatism

▼ Modern Physiological support

Constructivism is supported by the fact that on a physiological level, “knowledge” is just neurons firing and that the sensory organs send the same kind of signals to the brain, just from different location.

▼ Adaptation

- Jean Piaget says that cognition is an adptative function

- In evolution theory, adaptation is a result of natural selection and the actual altering of genes. It is NOT an individual deer growing a taller neck to become a giraffe.
- Piaget's adaptive on the other hand, IS a bit like a deer growing its neck to become a giraffe
- Piaget says that a set of schemes will survive in the mind only if it's ***viable***. More specifically, the set of schemes is at ***equilibrium*** with the stimulus that the organism receives and with each other's existence.
- Events that cause the schemes to change are called ***pertubations***

▼ Assimilation and Accomodation

- Assimilation is the act of assigning an unknown object to a scheme. For example, a child adding a spoon to the "rattle scheme".
- Accomodation is changing the schemes when a pertrubation (diappointment) happens. This may even involve constructing a new scheme.
- A scheme is a collection of properties of the object. These can be passive properties like color, shape, etc. or active properties such as "If I shake this object, it makes a sound" which as action-response pairs.

▼ Requirements of an organism to form schemes

- Remembering and Retrieval of experiences
- A sense of similarity between experiences
- The tendency to establish recurrences (patterns).
- Some experiences being more favourable (pleasurable/satisfactory) than others.

▼ Instrumentality of schemes

There is two-fold usage of a scheme

▼ sensory-motor level / utilitarian instrumentality

The organism benefits from schemes such as "predator", "parent" ,etc.

▼ Reflective abstraction level / Epistemic instrumentality

Some (operative) schemes don't have immediate usage, but help the organism achieve a more coherent conceptual set of schemes. These schemes are for example, "foresight" , "planning", "thinking" .

▼ The "other"

Some schemes such as "thinking" , "feeling" , "planing" which a child doesn't label but identifies himself doing will come in use when it sees another person.

The child will impute (attach) many of these schemes to the other person since there's already a lot of similarity between him and the other person. This can extend to a lot of other schemes that are about cognitive functions.

Now, when the child makes a prediction and the other person makes the same prediction, the child's confidence is his prediction is much more than if the other person had not made the same prediction.

▼ Language and Communication

According to Shannon (yes, the information theory guy), language is basically code. It is meaningless if the the syntax/representation of the items involved in the code are not understood by the receiver. Basically, you can't talk Mandarian if you don't already have the syntax and semantics for Mandarian.

The way that the syntax and semantics are learnt is through associating words to objects and concepts. For example, showing the character '2' to a child and many objects that have the property of "two-ness" will make the child associate the character to the property.

Once enough syntax has been built up, information can be communicated through the syntax.

Ofc since the encoding of information at the sender's end and the decoding at the receiver's end aren't exactly the same, the information is often altered. Plus, now two individuals have the same set of schemes. For some, sex is assimilated to the scheme of "romance" and for others it isn't. So when a young girl says "I want to experience romance", it can often cause other children to be embarassed.

▼ Performace and Understanding

While a child reciting a definition of a term verbatim might be enough evidence for a pure behaviourist to say that the child understands the term, any sane person would disagree.

For the child to understand the term means that the child must be able to form a scheme for the term. This, we can't directly check through simple recitation.

Thus the hate for rote learning.

But of course, there are things that you MUST memorise, say, multiplication tables or the postulates of quantum mechanics. Some things are just not worth creating schemes for.

For a long time, it was thought (really, enforced) that language carries meaning. And thus, just telling the child enough times or making it read the text should be enough for him to understand the information in that text. This is of course bullshit. Chatur (from 3 idiots) didn't grow up to be a scientist, only a performer (trained, but with 0 understanding).

▼ Group learning

Since (in accordance to Piaget) the only way to gain understanding is by experiencing pertubances, to aid the process of learning, the children must be facing as many pertrubations as possible.

One of the most important sources of pertubations are interaction with others and learning of their constructed knowledge, their definitions, etc.

So ... group studies !

Plus, as any P2P application is self-sustained, so is group study, meaning less work for the teacher.

▼ Constraints

Rather than providing a description of what an object is, one can describe what the object is not.

This allows for more fluid usage of the schemes already built by the student.

Basically, if a child comes up with a solution which is wrong, don't just straight up say that the solution is wrong, but point to the step that is. That is the constraint, something that the student must avoid.

▼ Flaw

Constructivism cannot explain learning of raw facts. There are no pertubances, in fact, no expectations. And yet, memorisation is part of learning.

▼ Situating Constructionism

- Learning is most effective when learners are engaged in creation of shareable artifacts.
- The process of constructing these artifacts incentivises acquisition of knowledge.
- Sharing the artifacts allows feedback and recognition which help in learning and motivates it (essential for learning, according to constructivism) . This is the "situated" part in the situated constructionism.
- Bricolage : This is when you create something in a ground-up approach, "talking" to the thing, with lots of negotiation between you and the project.
- Soap culture math : This is a very unproffessional term, but what it encompasses is the the "thinking, dreaming, getting a new idea, dropping it or persisting with it, and seeing other people's work and their reaction to yours" that happens when you work on a publicably shareable project, much like how actual (math) experts work, and school kids don't (when being taught math).
- Style : Constructionism allows for various styles of doing things. You are supposed to be creative and try new things, much unlike the gray, fixed way of doing things in school. This uniqueness offers both flexibility for the maker (allowing him to do things his preffered way, rather than feeling stuck, doing something that seems meaningless) and motivation (the maker can add his own special touch to the product, thus owning it, being proud of it, have it critiqued by others and also see the effects of different things that other do and learn from it on a much faster rate than he would by either instructionism or practicing on his own)
- Constructivism also allows Epistemological pluralism (multiple natures of knowledge).
- "Previously, the more abstract forms of knowledge (such as algebra) were preffered in schools. Now, schools are moving towards a bit more of the constructionist type of learning". This point feels like complete bullshit to me. A tool that you can't use as a sign eventually is just not something you actually learnt. Ask a proper JEE aspirant if math is "abstract" and you'll know. Math is just as abstract as is language or programming or even art. Idk wtf this guy is talking about. You make a turtle more from one position to other and you think what you did was more

“applied” than me figuring out how to program 3D animations all with my “abstract” knowledge ? Just because my concepts were learnt as scientic concepts rather than spontaneous doesn’t mean they are incomplete. Yes, experience helps, but that isn’t the only way to go.
In the end, we are still trying to teach the *abstract* concept to the kids, but allowing them to reach an intermediate development stage (the “concrete” concepts for concepts like length, angles, etc.), thus putting them in a ZPD for the abstract concept. That does NOT mean that you are removing the dominance of “abstract” knowledge from the school curriculum.

▼ Situating Learning in Communities of Practice

▼ Cognition plus view

This is the view that cognitive science should be used and extended to the context of a social environment to describe how the community affects learning on a cognitive level.

Cognition itself can’t be explained as a result of a social process (failure to apply the participation metaphor) since a guy sitting under a tree and thinking is also doing cognition.

▼ Interpretive view

Situated cognition is basically evolution of knowledge due to social interactions. Things such as language, protocols, etc. which as conventions are a function of the community and change over time. They aren’t fixed truths, like in science.

Even feelings, identities and all the stuff we usually don’t call “thinking” is now a part of the process of evolution of the conventions (which is now essentially knowledge).

▼ Situated social practice (constituitive)

- Agent and world are inter-dependent
- Meaning of things is socially negotiated
- Learning is about the relations among people (is it ?)
- The world itself is socially constituted. Basically saying that even science is historically and socially embedded
- There is a dialectical (two-way) relation between the person and the word. Both shape each other.
- The situatedness (of your learning) is historical. That is to say, the part is a large part of the the situation you’ll find in a community.
- Even the researcher is (unfortunately) part of the social practice.
- Learning is constituted withing the practice itself, not just with that as context. This is participation metaphor.

▼ Legitimate Peripheral Participation

- When a peripheral participant goes through the social process of increasingly centripital participation, he builds an identity, as well as the skillset required for being a more central participant (oldtimer).
- The process of centripital participation can only happen through a legitimate access to ongoing community practice for the peripheral participant (newcomer)
- A master does not provide the apprentice actual knowledge or skills directly. All he has to provide is the legitimate access.
- In an apprenticeship, there are no tests. This is based the participation itself is thought to be the learning.
- There are limitations faced by apprenticeships such as
  - having to give resources to less inefficient people
  - politics of management
  - Exploitation of naive people (looking at startups..)
- Someone named Marshal studied butcher apprenticeships and found it to be basically exploitation, with no actual learning involved for the peripheral learner
- Someone named Jordan studied Yucatec Mayan midwife apprecticeships. The tradition of midwifery is handed down to Maya girls who absorb the practice and eventually become central participants. They don’t have to ask to be taught. Neither is there a formal education for this. It happens only through centripital participation.

- Someone named Cain studied that “Alcoholics Anonymous” community, which is basically a group aimed at rehab from drinking. The stuff a newcomer does in the community is mostly just slient and symbolic. Eventually, they learn the lingo and start to participat centripitaly, leading to them reaching the final stage (which is where they convince active drinkers to join the community.. quite like a religion).
- When a community reaches an equilibrium, the rate of oldtimers “retiring” and newcomers joining is equal

▼ Commoditisation

- When the product that a participant generates is not related to the participant or the community directly, but meant for exachange in a market, at that point we say that the practice is commodified.
- When the participant exchanges his value (skills) in exchange of resources not tied to the community, then the participant is commodified.
- This objectification of the person as a unit of value, as well as attributing the power that should belong to a person to instead the product (such as Coca Cola) leads to alienation, where the activity becomes a means to the end, rather than itself being the end.

▼ Communities of Practice

Let’s have be slightly more open to the participation metaphor and take this more as a speech and less as an analysis.

▼ Four premises

- We are social beings
- Knowledge is a matter of competence with respect to valued enterprises/skills (such as singing in tune, coding, etc.)
- Knowing is participation in pursuit of the enterprises/skills .
- Meaning is the ultimate product of learning. This is what active engagement with the world is to produce.

▼ Participation

This isn’t just engagement in certain activities with certain people of the community. (one who’s just doing the activity, with no intent or appreciation)

Instead, it’s being an active participant of the community (one who aims to learn the practices and propagate it) and also constructing an identity (self identity or social ?) in relation to the community.

▼ Meaning

Our individual and collective ability to experience our life and the world as “meaningful” (quite a circular definition..)

▼ Practice

Shared historical and social

- resources
- frameworks and
- perspectives

that can sustain mutual engagement in action.

Basically the infrastructure and setup needed to sustain the community.

▼ Community

Social configurations in which the enterprise is (degfined to be) worth pursuing.

For example, stretching in a gym isn’t competence, but in a yoga class, it is.

▼ Belonging

This is different from identity. It also isn’t as simple as engagement + alignment.
It is about social relationships. It is also internal, just like identity, and not just how others see you.

▼ Identity

Personal history of becoming in the context of the community.

Note that you can have identity without having belonging through sheer competence and practice.

▼ Components of learning

Learning can be viewed under different lenses with different structures associated with it.

- **Learning as belonging** is when learning is done inside a **community**. For example, the Mayan midwife system
- **Learning as becoming** is when the **identity** is being formed. When a person says “I’m learning to be a husband”, it’s not about being competent as a husband, but identifying oneself as a husband.
- **Learning as experience** is when it’s tied to **meaning**. One could work very well in a job he doesn’t like and be very good at it, but still say “I learnt nothing”. And another could just scratch the surface of something that he wants to do in an underpaid role (looking at the poor PhDs) and say “I’m learning so much”
- **Learning as doing** is when learning is tied to **practice**. One says “I’ve learnt Python” or “I’ve learnt how to be a therapist” only when he has used it in projects, and not when he has only read about it.

▼ Family

Enterprise : Surviving together (actual survival or the survival of a viable identity in society)

The unsaid rules and equilibrium in a family is a social and historical thing. It’s the practice. If a new person (say a daughter in law) were to enter the family, she would have to learn all these unsaid rules. She’ll also make a few modifications of her own.

Of course if she doesn’t go through centripital participation, she may never truly be “part of family”. This can and does often lead to feeling of alienation.

▼ Office

Enterprise : The actual work.

Here too, while the company might set some rules, the actual rules are determined by the workers. If they decide that shifting timings from “9 to 5” to “10 to 6” makes sense (as a group) then eventually that will happen (looking at you Texas instruments..)

The office is very much a community of practice with many *masters* , *apprentices* and peripheral and central participants.

When the employee isn’t able to make any meaningful contribution and isn’t learning (going through centripital participation), they feel alienated.

(True. I’ve felt that once).

▼ School

Enterprise : Surviving school

Although the actual setters of the institution might not intend for a community to form on this basis, it does happen.

And often the most identity transforming learning happens in the communities that form in the cracks of the official structure of the institute.

Often, there are actually multiple communities of practice in a school : the band, the sports club, the cubing club, the back-benchers, the nerds, etc.

All of these communities combined is what we call “school”, although the formal definition only includes one community from all of it, namely the classroom.

A school could also be labelled better as an “organisation” rather than a community.

When someone is unable to participate in any community, he or she may feel alienated.

▼ Implications of participation metaphor

- For the individual, learning is now about engaging in and contributing to the practice of their communities.
- For the communities, learning (on the scale of the full community) is now an issue of ensuring new generation of members as well as refining their practice.
- For organisations, the issue is sustaining communities of interconnected practice so that the enterprise is still valuable.

▼ Theories of social structure

This is a very game theoretic view of communities. Here you model the actions of the (and in the) community as a result of its structure.

In an extreme case, you completely abstract away the individual and don't even care about it.

▼ Theories of situated experience

This is an extreme focus on the individual, dissolving everything social and "external" and only thinking of it as experience.

▼ Theories of social practice

This is when you are concerned with reproduction and production of specific ways of engaging with the world.

▼ Theories of identity

This is about the social formation of a person. Rituals, symbolicism, categorisation of individuals, etc.

▼ Theories of subjectivity

Unlike the theory of situated experience, agency is also a socially constructed thing here. Depending on the society one grew up in, his perception of things is bound to vary. That's where simple society independent agency fails. Even the sense of morality one has isn't inherent.

▼ Theories of power

This is about the influences you have in the components of the community itself, and not just on the practice (such as the normal propagation of community practice) . Essentially, you are in power if you can influence others in the community or make decisions on behalf of them.

▼ Theories of meaning

These are different from the theories of meaning in the philosophy of language or logic.

This is about how one's participation in a community leads to "meaningful" work for him, as a person.

▼ Claim Processors

(Legal) Claim Processing is a very boring and linear job from an outside POV. And the employees themselves also wish to just go home rather than working.

Moreover the work is almost impossible to do. It involves standing in queues to get your turn to operate on a computer, looking through documents and other mundane and time consuming stuff.

It shouldn't be possible for claim processors to work efficiently.

Yet, there is a very large informal community. The claim processors,

- share information and tricks
- help newcomers
- support each other emotionally during work through rituals (say tea breaks), dramas, etc.
- Provide resolutions to conflicts with the stupid institution (for example someone getting fired due to excessive phone time, even though the phone is literally the only way to do actual claim processing).
- Work as a communal memory that allows people to function without having to know everything.

The (informal) structure that the claim processors have developed in order to do their job is the *practice* . It is NOT the job itself.

▼ Acquisition and Construction based communities

- A community that only provides an access to competence in practice for the newcomers is a locus for acquisition of knowledge. For example, industry or academia.
- A community that incorporates new ideas, allows for changes in the status quo, and other things is a locus for the creation of knowledge. For example, a small team, or a family, or a research group.

▼ Strengthening of identity

To help the indeviduals have a sense of identity in the community, the community can have these policies:

- Incorporation of the member's past into its history. For example, the Romans adopting the Greek culture.
- Having opportunities of participation incentivised by a context of a valued future. For example, internships.

▼ Multimembership

This is when a person is part of so many communities that a coherent sense of identity is hard to have.

You see this in our language too .. “Work life balance” , implicitely saying that work and life are mutually exclusive.

There are communities interact and negotiate so that they may not be mutually exclusive.

▼ Peripherality and Marginality

- peripheral wisdom is the wisdom one has due to (not despite of) being a peripheral participant in a community; for example, a computer science student in a cog. psych. classroom.
- marginalities of competence is when a particular person is never considered a full paticipant (for example, a MTech student doing better research than most PhDs is still not counted as a “scholar”)
- marginalities of experience is when some experiences are not fully considered practice of the community; for example, doing qualitative studies in a quantitative field (say, cog. sci.)

Turning marginality into peripheral wisdom does not risk the core of the community. Although it may stretch the practice so wide and thin that the community may not even be recognisable anymore, giving rise to smaller (but better configured) communities arising from within.

▼ Reconfiguring identity

The community can reconfigure itself in two ways to provide more satisfactory sense of identity to the participants

- Internally; changing policies, labels, funding, etc. within the community itself.
- Externally; changing the community’s participation (yes, I know.. quite a statement) in other larger communities and the situation it finds itself in general. For example, CBSE schools teaching Python to kids at an early (objectively, no) age to make them better suited for the real world.

▼ Potential,Active,Latent

- A Potential community is a group of people who are not connected but share the same goals and experiences. For example, a community of people who are sacred of internet. There is a potential for the community to form, but since they are scared of internet, they probably won’t connect
- A latent community is one where the people do know each other, but there isn’t much interconnectedness. Consider for example, an extremely competitive classroom (thinking of my CS classes..). People know each other, even recognise each other as being part of the same community, but don’t really interact with each other.
- Active community:

  A normal, healthy community.

▼ Modes of participation

- engagement : The actual practice, striped of all the fancy decorations
- alignment : Following the standards and conventions
- imagination : Imagining yourself being in the role is an integral part of centripital participation.

▼ Cognitive Apprenticeship

- A cognitive apprenticeship is just an apprenticeship for learning the cognitive skills needed for the practice, as opposed to the physical or memory based ones.

▼ Types of knowledge

In a cognitive apprenticeship, the types of knowledge developed can be

- Domain knowledge : The actual concepts
- Heuristic strategies : Tricks of the trade that sometimes work, sometimes don’t
- Control strategies : This is the flow of the process; stages, decision making process for transitions between stages, etc.
- Learning strategies : These are strategies that the student learns in order to learn other things faster. For example, breaking a thing to its constituents, when to attach examples, when to do a google search, for how long to keep trying to solve a question, etc.

▼ Traditional methods of learning in an apprenticeship

Note that we are talking about process of learning, not the progression.

- Modeling : The master performs a task and the student observes and models his cognitive processes after the master
- Coaching : Observing the students and giving hints, challenges, feedback, and *scaffolding*
- Scaffolding : The support that is provided to a student to carry out a task (it's like side wheels on a bicycle. It's narrower than coaching which involves challenged and feedback too.

These are the methods is any apprenticeship. The next 3 are only for cognitive apprenticeships.

- Articulation : Making the student explain what he knows to extract his mental model of the process.
  Inquiry teaching is strategy of questioning the student so that they articulate their understanding.
  While articulation helps the master see the flaws, it also directly helps the student refine his understanding.
- Reflection : Making the student compare his process to an expert's process.
  This may seem similar to modeling, but is slightly different, since the student already has an existing articulated process. He only needs to change points of failure by observing what the master does.
  To get a replayable articulation fast, one can use video or audio recordings. This is why kids often learn better through videos, like of KhanAcademy rather than in a live classroom. Reflection works using incremental changes.
- Exploration : Allowing the students to try solving the problem by themselves, and even creating problem sets.
  For example, you may ask a student to create problems based on un-mastered topics.

▼ Sequencing (learning progression)

- Increasing complexity : Gradually increasing the depth of the task, so that more and more of the skills are used
- Increasing diversity : Creating problems that involve multiple skillsets (isn't this just increasing complexity ?)
- Global before local skills : Before going to the individual parts, it's better to do dry-runs of the full process.
  For example, in tailoring, the apprentice often put togather a garment from pre-cut pieces before they learn to cut the pieces themselves.
  This allows them to see the full picture and have the incentive to learn how to do the smaller parts.

▼ Sociology

- Situated learning : Have the students carry out the tasks in the same environment that experts find the tasks in.
  No miniaturisation, except that the tasks will have less risk associated with them and will be on a smaller scale.
- Community of practice
- Intrinsic motivation : Situated learning and creation of a CoP are ways to generate intrinsic motivation. Extrinsic motivation is about gaining something material; say, a grade. While intrinsic motivation runs on identity, belonging, meaning and so on.
- Exploiting cooperation : Pair up the students and make one of them inquire and reflect while other articulates.
  This is both low-cost and a good motivator.

▼ Scaffolding

There are usually 2 types

- Stucturing : Reducing unnecessary setup of a task so that the stages are already laid out, some of them even partially completed.
- Problematisizing : Increasing the task's complexity by piggy-backing additonal reflective tasks on top of the main task or doing inquiries at certain stages.

The idea is to keep the student perpetually in a zone of proximal development (ZPD).

Computer aided scaffolding is immediate and private. That is a really big thing now because of these two properties.

▼ Community of Learners

This is a CoP with the practice being learning itself. A CoP needs these properties to be a CoL :

- diversity of expertise among its members
- shared objective of continually advancing collective knowledge
- an emphasis on learning how to learn

- mechanism for sharing what is learnt

It’s not necessary for each member to know everything that the community knows, only who knows what.

To facilitate a CoL , people came up with the “Fostering a Community of Learners” (FCL) model.

In this, there are 3 research cycles per year with each cycle begining with sharing material to beild a common knowledge base, from where the “research groups” diverge and do their “research”. There are regular cross-talk sessions between the groups. Finally, at the end of the cycle, the groups come together and complete the jigsaw puzzle of information needed to do the actual task.

▼ Reflection Assignment 4

I need to select a skill I’m good at and write a short (300 word) plan articulating how I would teach it to a friend using cognitive apprenticeship.

I’ll choose a rather practical thing. I was creating LLM agents at Texas Instruments for doing Text2SQL conversion. Abstracting away the code I wrote, the models I used, and thinking of the backend as an actual person, let me rephrase the “skill” as :

Writing a SQL query on behalf of another person, in the context of a database to do some analysis task.

▼ Modeling

I would break the full process into multiple interdependent stages, such as

- Interpreting the asker’s question and searching for tables and columns in the database that are important for creating a SQL query that solves this particular question
- Decomposing the question into simpler questions in the context of the tables selected in the first step and
- Creating parts of the query based on the decomposition
- Combining the parts into a single functioning query
- Testing the query to see if it works.

I’ll do the steps for many different queries and let the student observe. He’ll be able to somewhat model the functions involved in each stage.
In reality, I just hardcoded the stages in the architecture for the application.

▼ Reflection

I used this mode of learning a lot for the LLM agents. I would just give example queries to the agents and observe patterns in the output. Then I would compare that with what the ideal output be. I would iteratively change the system prompts of the agents little by little. I understand that this is essentially changing their knowledge directly, rather than having them do it, but the process is very much reflection.

▼ Exploration

Although I had not built a mechanism that allows the tool to get better and better with usage at figuring out how to do divergent thinking, but still, one part of the process; namely the schema searching uses divergent thinking and exploration. I let the LLM agent “play” with the database and the RAG pipeline, with the intention to learn the kinds of tables there are (much like what a real person would do) and eventually come to a conclusion on what’s the best set of tables to use. I found this iteration based agent to be far superior to the simple system prompt based and one-time RAG search based agents.

▼ Articulation

Since I have a dedicated stage to simply decomposing the task, the tool has to come up with a plan and later use that for the actual query. This is articulation without feedback. I also had an option for the user to provide feedback to the tool at that stage. So even though the user can’t see the internal cognitive (computational ?) structures of the tool, it can still see the understanding of the tool by having the plan articulated.

▼ Scaffolding

Initially, I didn’t have all the stages, and neither did I have the iterative loop that the tool now has. At that time, I was simply testing the effects of having different settings, system prompts, search iterations, feedback at different stages, etc.

This can potentially be thought of as scaffolding. As I slowly built the tool, the need for manually attacking the stages and the constant debugging and changing of output went away. The tool could now accurately generate the

query all on its own.

Perhaps rather than thinking of teaching a single agent, I should think of the evolution of the tool itself as learning.

After a long chat with ChatGPT, this is what my submission will be :

Teaching Activity: Translating Natural Language into SQL

The skill to learn will be conversion of natural language questions and tasks into SQL queries in the context of given database.

There will be a global-before-local phase which will be a demonstration of the full workflow; decomposition, SQL writing, testing, and refinement; so the learner sees the end-to-end process and there is context to the tasks given to him. The mode of learning in this phase is mostly **modelling** based on the expert's process.
There will then be then increasingly complex stages of the apprenticeship, with the final stage being the full practice. This progression is a form of scaffolding.

1. SQL Writing

   Learner writes SQL directly from very simple natural language queries with the table schemas that are needed, already provided.
   After learning this skill, the apprentice becomes an asset to the master since the simpler problems can be given to him. For a deeper analysis of problems, the apprentice may even contact the client directly. This pushes the learner into a Community of Practice from the start.

2. Decomposition + SQL

   Learner breaks the task into procedural steps before writing SQL. This stage enforces **articulation** of reasoning and **exploration**. This skill enables the apprentice to take on more complicated conversions than before.

3. Table search + decomposition + SQL

   This is essentially the full workflow, but the focus is mostly on learning how to search relevant tables from the full database.

4. Full Workflow

   Learner independently performs decomposition, table selection, SQL writing, testing and starts to interact with the clients directly, developing an identity. The Instructor minimises scaffolding and coaching.

Reflection will be used to facilitate learning at any stage. After each task, the learner gets to compare their queries and outputs with instructor-provided examples.

▼ Expansive Learning at Work

▼ Four central questions

- Who are the subjects of learning (definition, location)
- Why do they learn (incentives, values)
- What do they learn ? (contents and outcomes)
- How do they learn ? (processes)

▼ Activity Theory

- Humans don't interact with the world directly. Instead our actions are mediated by cultural tools, such as language, technology, symbols.
- Our actions are shaped by social structures (rules, roles, communities).
- The *subject* is the person or group doing an activity.
- The *object* is the goal or problem space being worked on.
- The mediating artifacts or tools help or allow the subject to connect to the object.
- A community then, is a bunch of people who all share the same object (again, a goal, or a problem space)
- Gen 1 claim : Mediation transforms behaviour.
  For example, counting with fingers rather than with numerals lead to different mental structures.
- Gen 1 claim (ZPD) : A student can do more with guidance than alone and the student's ability will eventually catch up to that scaffolded zone. Thus, learning is social.

- Gen 1 claim (cultural mediation) : Memory, reasoning and attention are transformed by cultural tools. For example, a child growing up in a literate society has different memory strategies than one in an oral society.
- Internalisation of a social practice is when an external tool becomes part of the internal methods. For example, I talk aloud things to guide and refine my knowledge (private speech) . Talking is a social practice that used to be external, but is now internal.
  Thus, each higher mental function appears twice, once as a social practice and then as a psychological practice.
- Behaviourists thought that stimulus-response conditioning is universal. But when mediation enters, the learning pathway is altered. For example, a person who creates a story out of a memory set is likely to retain it, while those who don't will forget. The mnemonic or story is the mediating sign/tool .

▼ Five principles of activity theory

▼ Activity system is a unit of analysis

Rather than just analysing a single participant in an activity or community, the whole system must be analysed.

For example, rather than analysing a single coder in a software development team, you should analyse the entire team, the coders (subjects), sprint goals (object), the community (managers, team, etc.) and division of labour.
The participants' activity is then a sub-system in the larger system. There is of course interdependence of the individual and community activity systems. Activity theory doesn't ignore this.

▼ Multi-voicedness

Every activity system will have multiple perspectives, interests and history. This creates conflict, but also innovation.

For example, Yeshit wanted me to focus more on writing code and building things when I was working on Text2SQL, while I wanted to do a deeper analysis and focus on scalability, putting my "research and analysis" skills to use. The middle path was probably the best one, but there was also conflict. This is a case of conflicting ways of approaching a problem, not just conflicting interests.

Similarly, an economist and a psychologist may model romantic relationships differently. The two views may be conflicting, or may be simply orthogonal and might lead to innovation.

Note that multi-voicedness happens because the participants have different histories and use different mediation for accomplishing a task.

▼ Historicity

Activity systems evolve over long historical time-frames. The current state isn't enough to understand the system. There are tensions, grudges, hopes, etc. involved for changes in the system. You can't just take a snapshot and say "this is it".

For example, Python's culture of readability stems from a frustration from verbose languages like Java. Without that history, Python is .. just another language. If you want to understand why people working on the language do what they do, just looking at the current scenario isn't enough.

Tools, rules and concepts are cultural mediators that accumulate over time. Take readability as a concept.

A C programmer might call this "good code"

```
def f(input_list:list[int])→list[int]:
    """
    Doubles the numbers in list
    """
    n = len(L)
    output_list = [0]*n
    for i in range(n):
        output_list[i] = 2*input_list[i]
    return output_list
```

,while this is called bad code :

```
f = lambda L : [2*x for x in L]
```

And yet, the second one is more Pythonic and encouraged often. The readability norm is a thing that Python community has carried for ages.

▼ Contradictions as drivers of change

When there are tensions within the system or a mismatch of the output that the community values vs the output the other communities value from this community, and what really works, these situations are called contradictions.

For example, before the massive improvement in tools like Perplexity, TI's very restrictive norms about usage of these tools (mainly due to security reasons) was OK. But now, it's an hindrance to faster development. The main contradiction is between the efficiency POV and the security POV.

Note that a conflict is like a symptom whereas a contradiction is a disease. For example, a conflict between a proffessor and a student can be cured with time, but the contradiction with wanting to pass (so as to secure a degree) and wanting them to primarily learn (so as to involve them in research) will still exist even if the student was a top scorer.

Often, a contradiction is a conflict between multiple mediation (perspective) pathways.

▼ Expansive cycles

TI's norms are restrictive, yes. But this contradiction is also the cause of innovations, such as in-house LLMs, custom agents, and a way to securely access AI agents from outside or a community scale.

This is shift in the object (the enterprise (coding vs designing)) itself. Moreover, this will cause the object to expand to also include AI based web development and other things.

For some contradictions, the solution is a rather *radical* expansion of the objective (think, sexual revolution expanding romantic relationships from marriage to all sorts of stuff).

Such expansions happen cyclically. After an expansion, the system (over)fits to the new structure, then becomes outdated eventually, and then again undergoes expansion.

In the end, any expansion is about creating a new mediating artifact. (In the Ti exampl, the custom LLM agents).

▼ Mind and Society (chp. 3,4,5,6)

- Human perception and attention aren't just biological (natural) functions.
- Perception and attention are culturally mediated, whether we like it or not.
- The objects used in cultural mediation that a child internalises to direct his own attention and perception is called an auxiliary mean. (For example, pointing is an auxiliary mean to direct attention, but the whole process is just "mediation". Without knowing the mediation, the auxiliary mean is rather arbitrary and won't be internalised.)
- The natural form of perception/attention is an immediate, unchanging response to a direct sensory input.
- Higher forms of perception/attention (ones that humans are extremely good at) are often done using auxiliary means. For example, a toddler isn't able to focus on a particular task for a long time and is captivated by every distraction around it, but a student can cut down all the distractions, not naturally, but by *reminding* himself to pay attention. This is possible only using language and the experience of expectations/punishment/etc. that the student has internalised. In this sense, the student's capabilities are increased since he can "trick" himself.
- Mediation transforms perception/attention from being automatic (monkey see, monkey do) to voluntary (humans don't see (the goal/threat) but still do (by tricking themselves)) .
- When children were asked to sort objects, younger kids relied on the visual similarity (putting an egg and a white ball in the same basket) while older kids relied on verbal categories (putting eggs and bread in same basket, though they look different) .
- When kids were told to search for a red square in a picture containing lots of different objects, younger kids were reiterating the command "find the red square" physically, while older kids were not, since they had internalised language based mediation and were using it as a purely psychological tool, rather than externalising it.
- Similarly, memory too can be mediated.
- Natural memory is remembering by sheer association.

Higher memory is remembering using tools, such as creating mnemonics or stories, grouping things together, structuring them, etc.

- When children were given a memory set and external aids, such as arranging objects and creating categories they performed better than those who didn't get the aid.
- Tools are things that change the external world (hammer, nail, rearranging, drawing, speech),
- Signs are things that change the internal world (language, diagrams, notations).
- In the process of internalisation, the kid first uses external aids (tools) to extend his cognitive capabilities; in collaboration with others (hence cultural).
  Over time, the kid uses signs and uses them instead.
- Often, "development" of a child mentally, isn't actually his brain getting stronger, but rather him acquiring and internalising more of these tools, and using them more efficiently.
- Spontaneous concepts arise from daily experience.
  These are intuitive, situational and concrete.
  For example, a child learns of the concept "brother" just by interacting with his family. He may also learn of "sibling" and possibly think both words mean the same.
  A teenager learns "love" by just watching movies and being in relationships.
- Scientific concepts arise from systematic instruction.
  They are abstract, reflective and generalised.
  For example, child learns that "sibling" is an umbrella term for "brother" or "sister".
  Similarly, the child may not have a sister, but after being told what the word means, he would be able to construct the meaning of "sister" through the instruction.
- A Spontaneous concept is like training a neural net, whereas a scientific concept is like writing an algorithm.
- Spontaneous and scientific concepts both develop togather, not in isolation.
  Spontaneous concepts are the primitive content that a scientific concept builds structure out of.
  Just knowing what a sister is, is still inferior knowledge compared to having a sister. But suppose now that the parents have another child and the kid actually has a sister, he'll be able to learn the "full" concept of a sister much quicker than the time it took him to learn of "brother".
- Scientific concepts are usually learnt first through instruction and then become internalised and aid in re-organing spontaneous thinking. (Think, a teenage girl characterising flings as "love", when she doesn't know the concept of FWB, while a woman characterising it as FWB)
  Basically, using scientific concepts such as language, one can create metaphors, analogies and other things that were previously not possible. One can explore the properties of a concept much quicker than if he had no (or imprecise) scientific concepts related to that object.

  This creation of analogies, and structuring based on the primitive experiences and associations is what we usually call development.
  But if these are the only two ways of learning, then once can be taught a completely foreign concept only by slowly making them come close to it. For example, to teach differential equations to a kid, you need to slowly build up layers of terminology and techniques. This is done by acquisition of sponatneous knowledge (my teacher called $y' = x$ a DE, and same for $y'' = y$) and systemic knowledge (A first order ODE is one that has a derivative of first order) .

  Thus, at any given instant, the kid can only learn so much. For him to learn, there must be a bit of challenge, but not too much of challenge, i.e. he must be in a Zone of Proximal Development (ZPD)
- What exactly is learning and development ?
  - Maturationist view (Piaget) : A child must reach a certain stage of development before they can learn. For example, the child must be able to read and write properly before you can even start teaching algebra to him.
  - Behaviourist view : Learning IS development. Basically, any skill can be acquired by repeated practice. They think of kids similar to how we think of neural networks.
  - Vygotsky's view : Learning leads to development.
    Here, learning is a process, while development is an outcome.
    Learning (and eventually development) happens in the ZPD.

More precisely, learning is the process of acquisition and internalisation of new cultural tools, while development is the stage when the kid can use these tools to either make sense of the world or do some higher cognitive function using the sign (internalised tool) as mediation.

Learning is a collaborative process, while development in individual (and not a process). When we say “he is learning Mandarin”, we also say “from XYZ” . But when we say “His Mandarin has developed”, we don’t care about the source or process of learning; it’s all about the person’s capability.

▼ Reflection Assignment 5 | Trial 1

The situation

I was creating a compiler for a custom programming language in a compilers course I took and learning about compilers by doing this. It wasn’t the conventional classroom setting but rather a constructionist one. I had a teammate who was really knowledgeable in PL theory and the formal way of writing compilers.

Vygotsky’s Theory

The theory and techniques used in compiler design (lexing, parsing, code-gen) were the tools that had to be internalised. Now, whenever I need to think about creating a system that understands syntax of some kind (think, boolean logic on Scopus, or some shell utility), I use these internalised tools to design such a system.

Every week, I would build part of the compiler, and every weekend, me and my teammate would discuss my work and I would find what needs to be done next. Thus, I was always in the ZPD. I was slowly catching on to the PL and compiler design concepts.

Engestorm’s Theory

There was also a wider community in the course, namely the multiple teams all working on their own compilers. We (compiler developer apprentices) were the subject and designing/creating compilers was the object.

The system consisted of TAs assigned to each team to guide them as they worked, the teams themselves, the professor who designed lab activities and assignments based on the current progress of the teams (on average) to introduce them to mediatory concepts as progressed through the process.

Most of the teams just wanted a good grade and would take less risky paths, often setting lower goals for themselves. Meanwhile the professor and TAs wanted us to explore as many paradigms as we could. This was a fundamental contradiction in the system.

Personal Insight

What this experience taught me was that it’s not just my professors or teachers that I can learn from, but also my peers and coworkers.

▼ Reflective Assignment 5 | Trial 2

The situation

While reading about gases and thermodynamics, I stumbled across boltzman statistics and tried to derive it on my own (no classroom), eventually learning a few concepts from statistical mechanics.

Vygotsky’s Theory

I had never done a derivation on this scale before. I wasn’t able to do it with the methods I knew at that point. So, I search online, but found a lot of derivations using the canonical ensemble and partition functions, theory that was out of reach for me. Eventually, I found a derivation that was just in reach for me. That is to say, I was in the ZPD for that method. I read through it and tried applying it to other contexts, such as liquid solutions and electro-chemistry and found success in explaining things that I could only memorise before. At that point, I had internalised it.

Engerstom’s Theory

I had unknowingly entered the Statistical Mechanics community. Eventually I even went through a formal course at IITGN where I learnt the full range of techniques.

The subjects are the students, teachers, proffessors, researchers, with the object being solving problems using Stat. Mech. formulation.

Just like any academic community, the students usually only wish to get good grades, while the professors want the students to reach a stage where they can do research. I can imagine this being a fundamental contradiction, at least based on the attitude of my classmates in the formal course I took.

▼ Reflective Assignment 5 | Trial 3

The Situation

During my internship as a Data Engineering Intern, I had to work extensively with Oracle SQL. At first, I could only follow queries if there were detailed comments or documentation, but eventually I reached the point where I could parse complicated SQL queries just by reading them. This shift made SQL not only a tool I used but also a language I could understand as part of the team's work.

Vygotsky's Theory

This learning moment can be explained through the **zone of proximal development (ZPD)** and **mediation by cultural tools**. SQL itself was the cultural tool that mediated my interaction with the database, but I needed scaffolding at first: documentation, ChatGPT, and seniors. Over time, I internalised enough of SQL to work independently, treating it as a communicative resource rather than just executable code. (Note that SQL is just as much of a social construct as English or Mandarin, so it qualifies as a social tool).

Engestrom's Theory

In this activity system, the **subject** was me (the intern), the **object** was contributing to database projects, and the **tools** included SQL and supporting resources. The **community** involved developers, managers, and DB admins; the **rules** included security policies, coding conventions, and workplace routines like scrum meetings and WebEx communication; and the **division of labour** split responsibilities between managers (project allocation and communication), admins (maintaining infrastructure) and developers (adding features). A key contradiction was between efficiency (developers wanting to use external AI tools) and security (rules restricting them), which spurred innovations such as in-house LLM tools like the Text2SQL project I worked on.

Personal Insight (1–2 sentences)

Vygotsky's theory helped me make the most sense of this experience, because it highlighted how I moved from relying on scaffolding to independently understanding SQL as a communicative tool. It showed me that my growth came from social mediation as much as individual practice.

▼ Reflective Assignment 5 | Trial 4

Advice as a social tool

The situation

During my late school years, I began facing emotion-related challenges that my parents or teachers couldn't advice me on. At first, I relied on scaffolds like YouTube videos, Reddit posts and Philosophy quotes to self-regulate my emotions. Eventually, I started rehearsing this using an imaginary "wise other", and later, multiple imaginary characters with distinct personalities (similar to IFS) and now, I no longer need to verbalise my problems and advice or have imaginary conversations physically (through text or speech) and can advise myself mentally by fluidly switching between me and the "wise other".

Vygotsky's Theory

Advice itself is the mediating tool here, with the goal being self-regulation from an unproductive or hazardous emotional state. The advice I got from internet acted as scaffolding, allowing me to make less decisions on what to hear and what to do. I rehearsed this (similar to private speech for language) using journaling and imaginary advisers, and finally internalised it, and the advisory council along with it.

Engestrom's Theory

There is a larger system that I was part of. Namely, the self-help community on the internet. The subjects are people who are facing problems in their life and the object is to function in the best way possible in that context. The community consists of advice-givers and advice-receivers. They interact with each other under the rules of societal norms, sub-Reddit rules, some extra rules of the community itself, such as not joking in serious situation, or saying things like "It isn't that bad".

A key contradiction is the conflict in nature of advice coming from different philosophical lenses; such as acting virtuously and responsibly according to Stoicism, or acting out of spite and free will according to the existentialists.

Personal Insight

Vygotsky's theory made the most sense of my experience because the development stages that I went through were very similar to the stages that he describes. Engestrom's theory is less relevant here because I remained only a benefactor of the activity system, and not a contributor.

▼ Learning Pathways

Key Characteristics of Learning Pathways:

1. **Relational / Situated / Life-wide**
    - Pathways unfold across multiple contexts (home, school, community, informal spaces).
    - Learner agency manifests as brokerage, persistence, repurposing, or resisting within these contexts.
    - Social structures (privilege, marginalization) shape what is possible.
    - The learners can also reveal new pathways by taking them over the popular ones. This can be metaphorically seen as "transforming" the popular pathway for themselves by altering some actions. As a (local) path grows in popularity, it becomes another cultural pathway.
2. **Affective / Episodic / Life-long**
    - A learner's taken pathway develops over extended periods of time.
    - Single critical episodes can redirect or derail trajectories by influencing self-perception, motivation, or access to resources.
    - Personal identity of a learner is a result of all the practices (and thus the nodes; or the activity systems) that the learner went through in his lifetime. The (local) identity in a particular activity system or CoP is just a part of this bigger (personal/global) identity.
3. **Identity- and Motivation-driven / Life-deep**
    - Engagement is shaped by (local) identity, belonging, and competence in the practices that the current context (activity system) provides.
    - Extrinsic goals (practical outcomes, recognition, possibility of unlocking the next stage in a pathway) also drive learning.
    - Learning is intertwined with who the learner is and who they want to become. The learner weighs in the practices and external motivators offered by the future contexts/nodes on the pathway before choosing which context (from the available ones) to go to next.

Core Principles:

- Learning is **cultural**: it is enacted through participation in socially and culturally organized practices.
- Access to learning resources and experiences is **structured by social systems** (race, class, gender, ability, etc.).
- Pathways consist of **sequences of consequential participations** and transitions, not just isolated episodes.
- Learners' actions and positions are **both shaped by and shape** the social and material contexts they inhabit.
- **Identity, social positioning, and perceived competence** are central to progression along a pathway.
- **Structural constraints and supports** (family, peers, institutions) mediate participation and opportunities.
- Identity is
    - relational : motivated by the *social connections*
    - affective : changes how you *feel*
    - motivational : extrinsic or intrinsic
- The sociomaterial practices that are goal of learning are
    - Interests and concerns
    - Coordinated participation (seeing and interacting with others as peers)
    - Social Relationships (seeing others as *more* than peers)

    - Identity
- Sociomaterial arrangements are places, things, and infrastructure; such as labs, sports complex, etc. This is the first requirement for you to do any sociomaterial practice.
- Sociomaterial position is how other see you, things you are allowed to do, and so on.
- Sociomaterial actions are things that you do; your agency.
- *Prolepsis* is projecting on the future based on your past, and then using that projection, making a decision.
- Learning as participation is total increased participation, accounting for everything one does.

▼ Pedagogy of the Opressed

- Banking education is just a synonym for instructionism
- Problem-posing education is “education” in the original meaning of the word; to “bring out”. Here, the student is given problems to solve that causes questions and answers to these questions.
- Banking education only servers the teacher by making their existence have “sense”. In this model, the teacher holds un-questionable, absolute knowledge; that is deemed valuable (in the mind of the teacher) and the student is perceived to be devoid of knowledge and valueless. This is very similar to an oppressive regime.
- In Banking-education, there is a dichotomy of roles. You only have teachers-of-students and students-of-teachers
- In humanist/problem-posing/revolutionary education there are student-teachers (teachers whose knowledge can be questioned) and teacher-students (students who are capable of thinking and thus teaching).
- In the Banking-education, the teacher first selects the material worth cognizing over and then transmits it to the students with no concern about whether it is relevant to them, whether they understand it, or if it is correct in the first place.
- In Banking-education, the students are supposed to be passive containers. They shouldn’t question or have opinions of their own. This opression will eventually cause a revolution. The humanists refuse to wait for the revolution and want to do things by themselves.
- Those “revolutionaries” who rather than take an alternative mode of teaching, simply teach different content, basically as dogma are just propagating the Banking-education; just on the other side.

▼ Power and Socio-cultural Theories of Learning

- In individual cognitive theories of learning, once the context is used to create a mental structure, the context is irrelevant
- In sociocultural theories, cognition is inseparable from context
- Power is relational. It’s not “in the head”.
- form-function is same as tool-sign
- Sociocultural theories are different from individual cognitive theories because
    - A sign is actually a part of cognition, rather than just a cause for a mental structure (like a tool would be treated in the individual cognitive view).
    - Learning is studied in daily life situations, and not in the lab
    - As people change their contexts, their learning shifts. Thus, the context is relevant.
    - Learning is studied over time; not just for the given instance.
    - The process of learning is studied, not just outcomes
    - Power : As people participate they are constrained by others and exercise agency (brokerage for example) for themselves and others too.
- Mediation and Power
    - In Vigotsky’s theory, learning is the acquisition and internalisation of the cultural tool
    - The acquired tool is powerful in different contexts. (For example, LaTeX formated documents are powerful when publishing, which plain text or MarkDown is preffered for blogs.)
    - The cultural tool also becomes a part of cognition. So, it’s powerful in that sense too. Consider Darwin’s “survival of the fittest” being used as a basis for racial purity thoughts in minds of Hitler’s followers.

- Learning in Everyday Life.
    - Cultural tools have material and ideological histories and gain a residue of of the activity of prior generations.
    - There is no such thing as a decontextualised learning. Even the research method (interview, questionnaire, etc.) have a context for the participant.
    - Situated learning theory (CoP) says that learning is just increased participation. So, you *need* a context (by definition) in which a person learns (participates).
    - A person may have a lot of skills but not be seen as skilled by old timers. Thus in the context of the CoP, the person is not competent since he isn't recognised as competent, and thus; doesn't recognise himself as competent. Thus, learning is contextualised.
    - Identity is also contextual. Based on what context (work, gym, library) you are, you will behave differently. A particular identity (the projection of an identity on a particular practice) can be thought of as an "internalised self".. like a VM running on the real processor.
- Power in Everyday Learning
    - heriarchy of master and apprentice.
    - There is no inherent power dynamic in all CoPs, but some CoPs do have power dynamics; heirarchies, marginalisation, prioritisation, etc,
- Power and Unit of Analysis
    - In Activity Theory, power is codified in the rules and division of labour part.
    - Activity Theory can't be applied to broad systems of power. For example, racism as a system of power is not one single activity system.
- History
    - Development occurs on 4 timescales
        - phylogenesis (biological evolution)
        - cultural history (development of the cultural tools themselves) aka sociogenesis
        - ontogenesis (development on a lindividual over lifespan)
        - microgenesis (development of a particular concept, stage, etc.)
    - The 4 timescales are interdependent. The culture shapes a person, and the people shape the culture.
    - The best unit of analysis (according to Engestrom) to understand history is community or activity system (thus, cultural history timescale)
    - Limiting to just activity system, history is the study of evolution of
        - human objects (motives)
        - cultural practices that humans do to get their object.
- History and Power
    - Broad systems of power and oppression are embedded in the history of cultural tools (mediational means), which will eventually be embedded in cognition as signs.
- Genetic Method
    - Vigotsky's double stimulation method has a primary and secondary stimulus, and the goal of seeing the effect of the secondary stimulus (language) on the processing of the primary stimulus (experiences).
    - Another example of double stimulation is when a participant in an activity system are taught about the activity system framework by the researcher (secondary stimulus) and then asked to reflect upon videos and notes related to the activity system (primary stimulus)
- Agency
    - While sociogenesis and ontogenesis is an important deciding factor in how a person acts moment-to-moment, it doesn't decide it completely. There is still some agency.

▼ Co-Construction of Power and Learning

- Power is capacity to change or preserve these for individuals and groups

    - conditions
    - resources
    - opportunities
- Power exists at these levels
    - Macro (institutions, politics, culture)
    - Meso (schools, policies)
    - Micro (how individuals act and relate)
    - Embodied (behaviours, emotions, identities)
- Power acts in two modes
    - coercive (explicit) : rules, threats, force
    - ideological : status quo / normality / common sense
- Interaction analysis views knowledge as social and situated in material settings (cultural tools and stuff).

▼ Microgenetic analysis is

- cognitive and psychological
- on small timescales (minutes)

- Critical interaction analysis says that every tool used not only fascilitates learning but also enacts, perpetuates, and changes power dynamics.
- Moment-to-moment analysis is using real-time data such as recordings and videos for analysis rather than collecting data afterwards (through interviews).

▼ Since power in an activity system is codified in

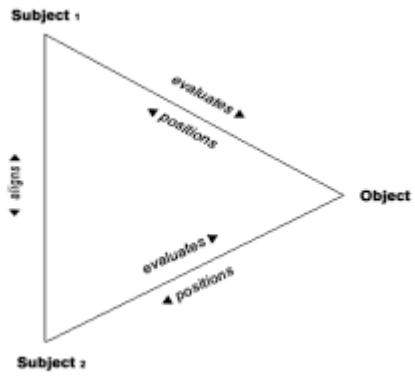
- rules and division of labor (norms of organisation)
- cultural tools (revolutionary/oppresive symbols)
- The subject itself (identity and ideology (constructivism))

thus, the papers read in literature review also had similar objects of study.

Namely, they studied from these views :

- dynamic construction of identity and ideology (Du Bois' triangle / Activity system + constructivism)
- organisation of learning environment (Cultural Pathways)
- leveraging and repurposing tools (Vigotsky's theory)

▼ Du Bois' stance traingle



This is in context of a discussion between two people (subjects) on a topic (object)

- The position of each subject w.r.t the object is decided based on the object. Example positions are "I feel this is good" , "I think it should change", "I know this is bad", "I have seen this before and it doesn't look good" , "I have a PhD in this and ..", etc. It's based on how the subject feels about the object and what the object means to him.
- The evaluation is what the subject says about the object, taking a stance/position in the process.

    - The other subject either does a convergent alignment with the first subject's stance or a divergent alignment (misalignment). This is based on:
        - The other subject's evaluation
        - The relationship (power dynamic for example) between this and the first subject.
- Du Boi's stance triangle is actually describing a part of an activity system. The evaluation will actually happen using cultural tools. The positioning might happen in the context of a community and an objective and the alignment might have to do with rules and norms.
- Micro-identities are temporary identities taken up by subjects during short interactions (acting as a slave in front of a proff. vs acting as a dictator in front of a slow teammate)
- Micro-figured words are small instances (say, telling someone to "man up" if they are lagging behind) of broader cultural tools (say, progress and hard work being a masculine concept)
- Ideological convergence is when a subject aligns himself with a position taken by a dominant "other" (another person, group, system, tool, etc.). This can either be
    - surface level compliance (only momentarily saying what they want to hear, but not believing in it)
    - internalised belief (a reconstruction of the position in the subject's mind)

      We are interested in the second kind of convergence. This is similar to assimilation, but in a distributed cognition setting.
- An ideological expansion is similar to accomodation (in a distributed cognition setting). It doesn't actually delete the old positions, but allows the subjects to have more positions and micro-identities than before.
  On a macro scale, it leads to expansive cycles.
- Representational tools are those used for higher forms, while discursive tools are ones used for interaction and power play (fancy scientific jargon for example).
- The representational tool (or any tool in general) can have ideologies embedded in it (Markdown not being accepted in a research community while Latex is, implicitely saying "we are not coders")

▼ To shift power, participants in a system can decide to

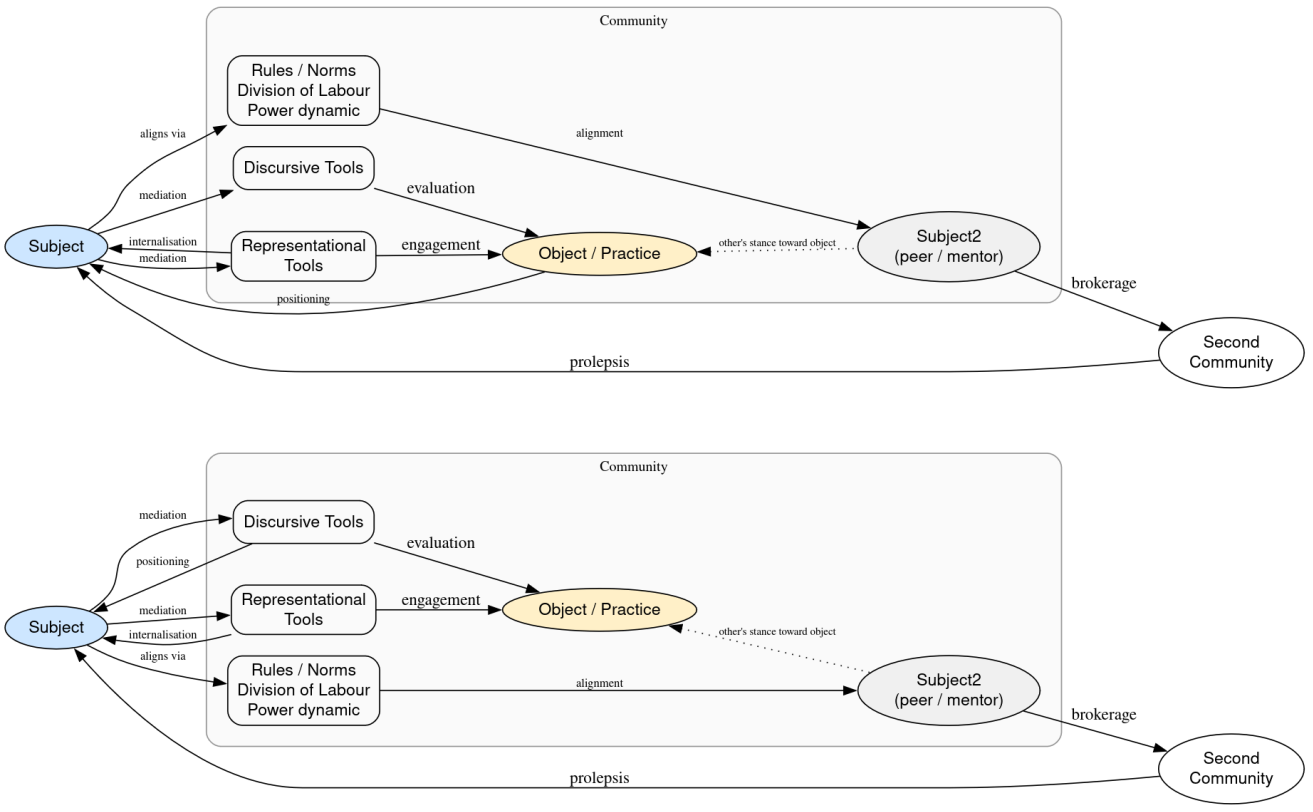
- change a cultural tool's meaning and *re-purpose* it (The LGBTQ+ community and their pronouns, or being in control of your emotions labelled as "toxic masculinity") or
- *co-create* (create a tool that co-exists with the old tool) a better tool (using "immature" rather than "womanly").

▼ A diagram

▼ code

```
digraph ActivityWithStance {
  rankdir=LR;
  fontsize=12;
  node [fontname="Helvetica"];

  /* Subjects and object */
  Subject [label="Subject", shape=ellipse, style=filled, fillcolor="#cfe8ff", width=1.2];
  Subject2 [label="Subject2\n(peer / mentor)", shape=ellipse, style=filled, fillcolor="#f2f2f2", width=1.4];
  Object [label="Object / Practice", shape=ellipse, style=filled, fillcolor="#fff2cc", width=2];

  /* Tools and rules (will be placed inside Community cluster) */
  RepTools [label="Representational\nTools", shape=rect, style=rounded];
  DiscTools [label="Discursive Tools", shape=rect, style=rounded];
  Rules [label="Rules / Norms\nDivision of Labour\nPower dynamic", shape=rect style=rounded];

  /* Community cluster (contains tools, rules, Subject2, division of labor) */
  subgraph cluster_community {
    label="Community";
    labelloc=t;
    color="#999999";
```

```
  style="rounded,filled";
  fillcolor="#fafafa";
  margin=18;

  /* order for nicer layout within cluster */
  { rank = same; RepTools; DiscTools; Rules; }
  Subject2;
  Object
}

/* A Community label node to visually connect / 'touch' Subject (dashed connector) */
CommunityLabel [label="", shape=none, width=0, height=0];

/* Edges representing the redesigned activity system with stance triangle */
/* Engagement arrow passes through representational tools */
Subject → RepTools [label="mediation", fontsize=10];
RepTools → Subject [label="internalisation", fontsize=10];
RepTools → Object [label="engagement", arrowhead=normal];

  /* Evaluation / positioning passing through discursive tools */
  Subject → DiscTools [label="mediation", fontsize=10];
  Subject → Object [label="positioning", fontsize=10 dir=back];
  DiscTools → Object [label="evaluation", arrowhead=normal];

  /* Subject2 aligns via rules (alignment arrow passing through rules) */
  Rules → Subject2 [label="alignment", fontsize=10];
  Subject → Rules [label="aligns via", fontsize=10];

  /* Optional direct interplay (forming stance triangle) */
  Object → Subject2 [label="other's stance toward object", style=dotted, fontsize=9, dir=back];
  /* A little spacing control */
  RepTools → DiscTools [style=invis];

  Community2 [label="Second\nCommunity"];
  Subject → Community2 [label=prolepsis dir=back]
  Subject2 → Community2 [label=brokerage];
}
```



▼ Preparation Assignment Week 8

Part 1 : Banking education and Problem-posing education.

The high school that I went to follows the CBSE curriculum and was centred in a rural locality where most students struggled communicating in English. I highly suspect this was because they didn't had access or affinity for movies, books, video-games, etc. wih English as the primary language for dialogue, sub-titles, instructions, etc., like most of the students living in urban areas had. This made it very easy for them to not be able to follow along; and easier still to not be able to question or discuss with teachers or authority.

The primary justification that any teacher would use to silence a kid would be their bad grades; which because of the situation I just mentioned were already rather prevalent. Most kids would identify themselves as "just not capable of it" as the teachers had been telling them from the start. Hence the dichotomy between teachers-of-students and studens-of-teachers always prevailed. The teachers could get away with poor teaching and worse behaviour with the students.

The dichotomy was in-fact a cultural artifact along with other dichotomies as a mature/knowledgeable elder and immature/un-knowledgeable kid. This structure was (and is) self-preserving.

I was transferred to this school from one where the curriculum (and the workload) was much tougher. I found it extremely easy to thrive in this easier environment with much less competition. I quickly became a "bright" student in the eyes of my teachers. Moreover I had access at my home to all the things that would foster my real education. I was also very interested in science at that time (thanks to many of the pop-sci channels like *Discovery*, *National Geographic*, etc.) and proficient in arithmetic (thanks to my father who would quiz me *a lot* and the older school I mentioned) . I would read books ahead of time purely out of curiosity. I had also found another student; who just like me; was very interested in science. We became rivals and best friends quickly. It was the most fun I had ever had in my life. For the first time in my life I was part of healthy competition. The fact that we were also at the top of our class consistently just made the rivalry even more important. It helped me improve me in every way possible. I often identified with him.

This backstory serves to explain how and why I was an outlier. And during my 9th and 10th grade I started to become aware of it.

During my 9th and 10th standard I could also see the pointlessness of many of the practices that the institution enforced. I will highlight some :

- Regular morning assembly where we had to sing (Hindu) prayers (I was an atheist at that age and it didn't make sense to me; some other kids were also from non-Hindu backgrounds) and a small group would perform in front of everyone things such as recitation of multiplication tables, news (I wasn't of the age to get interested in politics or news in general) or poems (which were in English and incomprehensible to most students).
- Students having to ask for simple things such as going to the washroom or drinking water (despite the bottle being in the same room).
- Students having to ask for permission to come inside an active classroom; most of the times doing a weird sign with their hands which in hindsight looks a *lot* like the Nazi salute.
- Students having to wear their (uncomfortable) uniforms with ties and belts tightened to suffocation.
- Boys having to have a particular length of hair and girls having only a particular amount of pony tails (I don't remember this part very well).

Add to all this the fact that there was humiliation and bullying of the students by teachers (and of course; other students) with any sort of failure to comply with the code being the trigger point.

Add to that the fact that most teachers were very religious and would add their opinions related to that in any topic; and I was an atheist.

Whenever I would ask why were the practices the way they are, I would get the answer "we are trying to teach you discipline".

One should not mix the words obedience and discipline so lightly. What the institution was really trying to teach me was obedience to a status quo that is a historical artifact from the military schooling tradition and shouldn't be allowed to demoralise and alienate the students from the practice of studying. There was no formation of identity in relation to the practice of studying for most students. Erik Erickson writes that children try to identify not just with a practice, but the masters of that practice. If the children don't respect the masters then there will be very minimal identification. And truthfully, apart from a few who were excelling on their own, most students had dis-identified already at that point. The absence of extrinsic motivation such as societal pressure (it was a rural society with not much cultural focus on education) also didn't help the situation.

Just as Friere writes, eventually the pressure that the institution puts on the students led to a “revolution”; led by no other than me.

The making things that I questioned were the norms and practices that the institution operates on, in the name of “discipline”. I started with asking questions on these topics to the more oppressive teachers in front of the classroom, while keeping a pretence of innocence. And since the teachers viewed me as a “bright” (obedient) student, thought it really was just innocence. Next I started not obeying the “discipline” and making it rather explicit. Initially there was friction; but with enough persistence the enforcers gave up. I had broken free of the banking model. I was free to choose a form of discipline that actually suited me. In my 10th grade, I took the first step of taking the matter of my education in my own hands. I started reading books ment for 11th and 12th grade; particularly mathematics. I was already going to a coaching institute at that point where I was being taught 11th grade science. I started valueing myself based on my scores on the weekly tests of this coaching institute.

I have already mentioned that I was at the top of my class. This was most of the times, without any real effort. I had made it a point of pride at that point and also a basis of my belief that I (and by extension, anyone) can learn anything given time and correct situation. I started teaching others in 10th grade. I tutored another student for the final centralised examinations held at the end at 10th grade. He was perceived to be a poor student by the whole class, but in merely a month he had become much better. I wasn't even all that good of a teacher. I believe it was because of his own effort. As Friere puts, I was a teacher-student.

In contrast, the education system at IITGN; the only other formally recognised institute in my life (I had a “dummy” junior college) is very much a problem-posing education. I've found my peers and professors often intaking and supporting the knowledge and ideas that originated from me rather than other way around. Some examples are

- I generalised a problem that was presented to us in my Discrete Mathematics class and found a solution on my own.
- I helped out my peers in writing the code generation part in my Compilers class.
- I suggested my OS proff on how the course could have been held better and he appreciated it.
- I brought in a lot of signal processing (convolution theorem, CLT, etc.) knowledge and information theory knowledge (entropy, cross entropy, etc.) in the Statistical Mechanics course I took.
- I created a new methodology to estimate impact of changing a software module in my Software Tools and Techniques course.
- I wrote a e-print on a new method for estimating Jaccard containment via sampling. A peer of mine found this e-print and read it for his literature survey for a research project. Even the Biostatistics professor I showed it to found it impressive and offered me a role in an industrial project.

There are of-course instances where power dynamics influence behaviour but these dynamics are implicit. The micro-identities I take on are because of my respect and belonging towards a person and how much I identify with them. That is, the power is ideological in nature rather than coercive (not counting the fickle process of grading). While I might enact a particular micro-identity when it comes to grades and crib sessions about marks on exams; I know full well that it's just a means to an end and not a part of my actual personality, and the professors know it too. Once the crib session is over, everyone goes back to their default identity (projected on the practice and context).

As mentioned before, during my 10th grade I was actively fighting against the oppression that the institution was making everyone go through. This started because I started noticing differences between how I acted at home and how I acted at school. A lot of it stemmed from the media I consumed; the movies such as “3 idiots”, “Taare Zameen Par”, etc. all being one factor. My inclination for science and eventually atheism (inspired by people such as Stephen Hawkings) was another factor since it forced me to take a stand for my own beliefs when being scolded by my religious father about it and thus make me more comfortable with standing up to oppression in general. The fact that a lot of practices followed in school were also religious only helped me dis-identify from the identity of an “obedient and cherished student”. Instead, I started to identify with the revolutionaries (freedom-fighters) that I saw movies about and geniuses like Einstein who had also condemned the schooling system.

Most kids believed that the system is absolute and it’s useless to try to change it in their favour; and had became so habituated/aligned to it that they would take the stance/position (like in Du Bois’ triangle) of the oppressors. The alignment was easy since they had no other reference subjects to align or identify towards. I on the other hand, *did*. This is very similar to indoctrination of kids into religions or cults.

Part 2 : Praxis

To take the most recent example, in this class (learning sciences) I had an issue with the way that words “concrete” and “abstract” were used in the readings for constructionism. The readings appears to link “concrete” with observable or

non-symbolic; something you can experience rather than deduce. This is not the definition that we settled on after the discussion in the class; namely that “concrete” is meaningful to the learner; something that can be understood and used in the contexts that the learner finds himself in day-to-day. For example, the fourier transform is an abstract knowledge to a zoologist, but not concrete knowledge to an electrical engineer who has to deal with signals a lot. Similarly, geometry, animation, and programming are concrete knowledge to kids who just want to have fun, create cool graphics, or small games using a software they have been introduced to; but manipulating symbols and finding values of variables from equations is not meaningful to the kids, and thus abstract knowledge.

We got to this definition not through simple instructionist transmission of the definition, but rather co-creation (or dare I say, *construct*ion) of this definition (knowledge) through dialogue and critical reflection (questioning the partitioning of knowledge as concrete or abstract).

The same thing happened when we were discussing the meaning of (no pun intended) “meaning” in a community of practice. We settled of defining meaning as an emergent property; just as one would define “romantic love” or “consciousness”, in contrast to what the lecture slides said (meaning is learning as experience) and the circular definition that was there in the reading material.

▼ Preparation Assignment Week 8 Summary

Freire’s “Banking” and “Problem-Posing” Education

My high school followed the CBSE curriculum and was located in a rural area where most students struggled with English. Because they lacked exposure to books, films, and games in English, they could not develop fluency. This made them perform poorly, which teachers then used as justification to silence them. A student with bad grades was automatically seen as incapable and undeserving of being heard. Teachers demanded obedience and called it “discipline.” Questioning was treated as defiance. This was Freire’s banking model in its clearest form: knowledge being deposited from above rather than discovered together.

I was an outlier. I had come from a tougher school, had access to English media, and was already deeply interested in science. These influences shaped my identity as modern, rational, and unorthodox. Seeing myself as scientific and atheist, I naturally opposed traditions and authority that seemed arbitrary or dogmatic. I began noticing contradictions in the institution’s practices, like forcing students to recite religious prayers or follow meaningless uniform rules. This recognition marked my conscientization. By tenth grade, I turned this awareness into praxis by questioning such norms in class, refusing unnecessary restrictions, and helping classmates who had been labeled weak. Their quick improvement showed me that the system had failed them, not that they had failed learning itself. Teaching them also deepened my own understanding; I became both teacher and student at once.

At IITGN, I finally experienced problem-posing education. Professors invited dialogue, and knowledge often moved in both directions. For example, during a discussion on “concrete” and “abstract” knowledge, our class concluded that concreteness depends on what feels personally meaningful to the learner rather than what is directly observable.

▼ Democracy and Education Chapter 11

- According to Dewey, any experience is always a part of a series of events and allows us to form patterns and predictions. A meaningful experience involves a stimulus and response pair. Thus, words cannot replace experience as those are only a symbolic description of the stimulus and response. (Basically trying to tell a color-blind person what red and blue are) .
- Dewey believes ounce of experience is better than a ton of theory. He does NOT say that scientific concepts are not knowledge and only spontaneous concepts are; but rather; spontaneous concepts are *better* .
- The three kinds of thinking are :
  - routine : Only focusing on the outcomes
  - capricious : Only focusing on the actions taken (that in return gave an outcome)
  - reflective : Trying to form a causal relationship between types of actions and outcomes.
- We are focused on the kids reflecting on topic X based on an activity A. If it instead reflects on topic Y, the experience (action A → response R → reflection Y on (A,R)) is not *productive* .
- Emboddied learning says that math and science is actually based on bodily function.
  - Stroop like experiments (primitive sensory stuff triggering language module unconsciously)
  - Meditation music

    - Private speech
- Thought begins in partiality and is concrete/instrumental and then becomes impartial and abstract/non-instrumental .
- According to Dewey, there is a dominance of something like cartesian dualism (body and mind are separate) where the senses are only inlets to the “mind” and “thinking” only happens in the mind.
- An activity that leads to no (or minimal) reflection is not an experience.
- An ancient Greek paradox is “You either know things or you don’t. Things that you know, you can never learn; and things that you don’t know, you can’t inquire about. Thus, you cannot learn without an experience being forced upon you”. Dewey points out that this is false since you can still makes hypothesis’ and test them. Thus, there is a way to progress knowledge even without experience (consider the fact that Einstein predicted black holes before we ever found one).
  Thus, Dewey himself is saying that scientific concepts exist and are important.
- Dewey also says that thinking is triggered only when there is a need; either direct (a war general having to plan) or indirect (people in a neutral state receiving news about the war and hypothesizing). So, without personal or inter-personal stake, one won’t think.
- Dewey says that thinking is an attempt at completing an incomplete picture.
    - The cognitive structures involved in thinking are instrumental to thinking rather than being the end goal. The end goal is the completion of the picture. Thinking, in a sense is an action one takes. It has the capacity to change future (duh..). Thus, thinking is also an experience (and a meaningful one since it is part of a bigger process).
    - Thinking is not *just* guessing. It also must involve testing the hypothesis or straight up proving it. No matter how likely something seems; as long as its probability hasn’t been impartially assessed and found to be sufficient, we cannot say that it is true.
    - The general features of a reflective experience are
        - doubt and confusion due to incomplete situation (trigger)
        - a conjectural anticipation (prior)
        - survey to gather information that might help in decision-making. (data collection)
        - a refining (inference) of the (posterior) hypothesis after seeing the information.
        - Projecting the hypothesis onto the set of actions available and taking a stand (prediction and response)

      The better that the data and inference is; the better the reflective experience.
    - Thinking is prospective (predictive) while knowledge is retrospective (data). Knowledge servers two purposes : first, its use in thinking and seconds because it assures us that the incomplete situation it arose from in now complete.

▼ Democracy and Education Chapter 12

- Acquired knowledge (learnt cognitive structures) and skill (internalised tools) that are not used (instrumental) in thinking are *dead* and thus does not help the person acquiring the skill.
- Even then, the dead knowledge does give a sense of completion and progress; basically it simulates real knowledge (the residue/certificate achieved after participation in thinking). Thus, it kills curiosity and the practice of real science.
  Such knowledge is termed as the “abstract” knowledge by constructionists.
- While shools do assign problems to solve; it doesn’t make them problem-posing instititutions. The problems that they assign are merely tests of recitation of abstract facts and methods. Genuine inquiry is not supported at school. The main issue is that the problems that a student tackles in schools are not *his* problems. Thus, he can never really have a reflective experience solving them. For that to happen, he must initiate the inquiry or hypothesis testing *on his own* or at least have the willingness to.
- Schools also implicitely teach kids that half-baked (incomplete), un-earned (spoon-fed) knowledge is acceptable. The kids then go through life without a lot of relflective experience since they are now ok with accepting dogma and propoganda as knowledge as long as it is coming from a “higher” authority.

▼ Assignment | Deweyan cycle

The way Dewey describes the stages of a reflective experience are basically how the research methodology was taught to me throughout my time so far at IITGN.

So, it makes sense for me to describe a research experience that I had.

▼ Phase 1 : Trigger

I was working as a Data Engineering Intern during the time that I started working on this question. My work was heavily centered around Oracle databases. I had to build a Text-to-SQL application that would use LLM agents to convert a natural language query to SQL (a programmatic querying language). This involved creating a retrieval pipeline for the different columns (of tables) in the database that matched the natural language query. Usually this process is done with the help of manually inserted descriptive metadata about each column. But that was not present for the database I was working on. So, I decided to mass-generate this metadata using features of the columns (data-type, length, number of unique elements, sample elements etc.) among which I kept a feature that I felt was crucial; which was the other columns in the database that were
"similar" to a given column. To define a measure of "similarity", I used the Simpson similarity score (which is basically the maximum of the two Jaccard containments) between any two columns. In theory, computing this was easy. But when I actually wrote the script and started this computation; it too too long. I did a quick complexity analysis to estimate the time it would take and found that it would be many days before the execution would finish, and this was on the test-database; not even the production database.
My plan was doomed; I thought.

▼ Part 2 : Hypothesis

But then, rather than simply giving up, I searched to see if there is a better method available.

I was taking an immense personal stake; both in terms of social relations as well as my own identity as a self proclaimed "problem solver". In the bigger context of building this application, I could very well have skipped working on this particular problem; and yet, I somehow ended up taking the harder path.

My hypothesis was rather simple: there exists a clever, easily implementable solution.

▼ Part 3 : Data collection.

I searched for 2 days straight, every kind of research paper related to this problem coming from all kinds of streams; data science, signal processing, probability theory, ecology, etc.

I implemented a lot of these solutions too and estimated (bottlneck and complexity analysis) the time running them would take.

None of them gave me a considerable speed-up.

Then, somehow (I don't remember why) while having a conversation with chatGPT, it have me a very simple way to model this process statistically. I went along and tested it too; and turns out it was working. That day I was in disbelief the whole time. I inquired about references and details and there were non. ChatGPT had **guessed** this solution. Of course, I had given it context, requirements, constraints and a lot of ideas; yet it figuring it out just like that and it not existing *anywhere* on the internet meant it must have been an obvious truth just hiding under the nose.

▼ Part 4 : Update of belief

Armed with this simple but new method I implemented the first sketch of my solution . I did some comparisons with other methods, and then wrote the final script. Even though it was slower than what I thought; I knew I am at the top of the iceberg for this method. I knew this could be improved significantly. I also suddenly had far more questions about the nature of the speedup it is giving .

By this time I was running out of time for the project and I had to compete other parts. But this idea would stay with me for future.

▼ Part 5 : Acting on it

I took all the methods that I had tested and after an ad-hoc analysis of the different regimes that they function best on, wrote a script that would run different methods for different regimes. I optimised the code, refactored it, binded it with the rest of the code-base and ran it to completion; which took about 2 nights. Finally, I asked my mentor whether I can write an e-print on my findings right before my internship was ending. After all this over; I actually started yet another research cycle, improved my analysis and solution and pushed it to ArXiV.

It sits there with the title "Sampling based Estimation of Jaccard containment and similarity". It even got included in the literature review that a team-mate of mine in a course I am taking this semester did (for a project of his own).

▼ Experiences in classroom

While a full-blown research cycle is infeasible and ineffective in a classroom; IITGN BTech. (especially CS) courses have a history of having a project component. This is mostly so that the students can apply their knowledge; but

sometimes it leads to acquisition of instrumental, concrete knowledge which has a lot of overlap with the syllabus.

One such experience I had was building a compiler for a custom programming language in the Compilers course held last year. I learnt all the core concepts of PL design simply through reading and designing. This was a very constructionist way of learning. The knowledge that I got from that course goes way beyond just facts and techniques; it was an **experience**; one that I enjoyed.

It might not be possible to teach everything through constructionism; but it certainly can be employed much more than how much schools employ it.

During my 12th grade, I was being taught thermodynamics, kinetic theory of gases, and electro-chemistry; all feilds relying heavily on statistical mechanics; and yet we were never taught statistical mechanics. Equations were memorised and applied, not derived. Different physical situations that were all prime examples of statistical ensembles were taught about separately. All this was because “it’s wasteful to study what’s not in syllabus” and the syllabus was the way it was because “the students won’t be able to handle the complexity of derivations”; and yet the students were believed to be able to handle mugging up these situations separately; all the while the studies that Vigotsky reviewed say that grouping helps in retrieval from long-term memory.

Since I was a “problem solver” I actually derived many of the complicated equations that others only memorised. While this had no effect on my performance (neither decrease nor increase) since I spent almost the same time with the material as everyone else; it did help me not be alienated from the knowledge. The way I did my derivations, they were similar to reflective experiences. Essentialy, I was re-inventing the wheel; which although is inefficient; is an integral part of learning in my opinion. It allows the learner to maintain continuity and instrumentality of knowledge and form a solid mass of concepts rather than separate clusters of concepts that are weakly remembered are need to be revisited.

It also removes the learner from the position of a consumer of knowledge to a creator of knowledge; just as Dewey wants learning to be.

Apart from all these constructionist experiences, I also had the prolonged experience of problem solving while preparing for the JEE examination.

▼ Dewey and JEE

The problems; apart from being tough; were often inspiring and thought provoking. Every fact and technique was a tool/sign that I had learnt that I used in my development (achievement of proficiency) and attacking problems. Problems were not about recital of knowledge, but rather application of it. You would usually need to cut-and-try multiple approaches for the same problem. You would guess these approached based on intuition; rather than a mechanical and deterministic method. Moreover; we would often just gloss over the material and jump straight into problem solving. Our teachers would constantly tell us that the main way to get better is to solve more problems. We would persist and often derive equations on the spot; effectively re-creating knowledge. For example, I never had to memorise the equations for projectile motion; I would simply derive it on the spot if needed. This allowed me to solve the varieties of projectile motion problems that required more than the simplistic, standardised approaches that were taught in schools and junior colleges. We would also try different heuristic and control strategies such as solving chemistry section first, or maths section first; going slow and accurate or fast but less accurate; persisting on a problem for 2 min or 5 min, drawing figures as the question is being read, converting to SI units first, writing the final analytic expression before plugging in numbers, etc. and add it to practice if it seemed to improve performance. Thus, apart from each question being a reflective experience; the process of improving performance was also a reflective experience. Coaching teachers would almost always give us 2 min to solve a question that they were going to model before actually modelling it in front of us. It was never just instructions that we were receiving; but rather a *solution* to *our* confusion.

▼ Instrumentality of JEE knowledge

Although there was direct instrumentality of the concepts; the concepts were also instrumental to other concepts (at least I made an effort to link it all) and the full mass of knowledge was like a massive network of these concepts.

One can question whether any part of this is relevant to the student’s “real” life; but I should remind you of the craze that is present in India about this exam: kids leave home at an age of 17 to *only* study in some unknown city in a state their culture has no matching characteristics with; spend their days *only* studying to the point of exhaustion and fatigue; commit suicides due to failure in this exam, and form online and offline cults with specific coaching teachers at the center. In the eyes of a JEE aspirant; their study *is* their life. Or better yet; it’s actually a matter of life and death to them. Getting one more mark could be the difference between being humiliated and having to re-do everything for yet another year (called a drop year) or being perceived as “smart” and on-the-way to great success (at least that’s how it is seen from the outside).

So, from this new perspective, if there is a way for the student to improve his score; a better, more efficient method; he **will** learn it. It's on-par with the sensory-motor instrumentality of schemes. Moreover, if there is a concept that could make it all "make sense" to the student; that could solve contradictions in his head and that could make the seemingly incoherent mass of techniques he is learning seem coherent, he will at least feel tempted to learn it; just like I did. This other incentive to acquire new techniques if the "reflective"-abstraction kind of instrumentality of the concept.

While it may seem what I am saying is not a valid solution, I beg you to view it more scientifically. Consider that we were to classify courses based on the frequency of quizzes and weightage to exams it has. I have found it through analysis of my peers' and my own patterns of course-taking that the "hard" courses; those with more quizzes; were often the more productive ones. It's not that we had more intrinsic incentive (motivation) to study for them; but that we were simply asked more questions and thus had more opportunities for learning.

Almost every performant exam-taker does one or two mock tests before the real tests. Their grades are usually high; but more than that, their curiosity is higher. They ask more questions in class, are confused more, and are often more creative.

While I do not have large scale data to back up my claim; in keeping with the spirit that Dewey is trying to instill in us students (and by extension in me), I should at least take a guess.

While one might say that these "motivated" students are a "different" case than the "normal populus", it would be spitting in the face of the cultural learning pathway. We *all* were once "normal". Heck, I was once just a conceited kid in a small town who believed he was the smartest person in the room. My ex.-g.f. was a normal girl overburdened due to her family who was using this exam as a gateway to a "good" university and escape from them. The top achiever in my batch was just an "average" kid at the start of first year. My room-mate in the first year was just an ambitious guy who wanted (and will be) an aerospace engineer. None of us were "extra-ordinary".

Coming back to my point; what I am trying to say is that with high enough stakes for exams that require more than just mugging up; that have unpredictable patterns and that test cognitive capacity in terms of speed and accuracy; and that are based on a network of knowledge rather than differentiated blobs; every concept becomes instrumental and thus concrete.

Once could argue that *anything* that anyone ever learns is done for an extrinsic goal such as money, social recognition, a continuation of identity and pride is in a sense. Learning out of pure curiosity is rare. Often the way that we solve problems are goal oriented; we don't focus on the concepts themselves as the end goal but rather as weapons to attack the real problem.

▼ Problem-first classrooms

In a problem-first classroom, learning begins with a challenging, authentic problem rather than pre-packaged theory. In my Compilers class, the teacher first showed us PLs designed by students in the previous year, and put creating such a PL as the final goal. When designing our own compilers, we proposed solutions, debated ideas, and experimented, while the teacher introduced concepts, symbols, or terminology to the class only when they became necessary to make progress. Each idea was instrumental in resolving doubt, following Dewey's reflective cycle of confusion, hypothesising, testing, and reconstructing understanding. Similarly, in my OS and Computer Networks classes, students were asked to critique naive solutions, identify bottlenecks or edge cases, and iteratively evolve systems, mirroring real-world design. JEE coaching also applies the same principle: concepts are taught just-in-time to solve problems, not as abstract goals, and each worksheet problem is a reflective experience in itself (guessing methods learnt from past problems to solve it and learning from the current trial). In short, in my problem-first classroom;

- The teacher first asks the students the *questions* that are solved the theory/technique they are about to learn in the next few lectures. This creates doubt and confusion in the mind.
- The *students propose/guess* either complete or partial solutions.
- The class and teacher go over different solutions proposed by students.
- The teacher highlights any flaws in the solution or any hidden nuance in the question that went un-noticed. The class has thus completed testing the proposed solution.

- The final solution that came from the discussion is compared to the actual solution given by experts. With a good discussion and a good enough teacher, these two are usually close. Finally the teacher *explains* the actual solution to the class.

This process is fairly easy to achieve for fields such as mathematics, physics, and especially computer science.

For more subjective fields, such as politics or humanities where an in-class discussion is much harder to direct to the correct end-point, the problem-first approach can instead be applied with the help of a personal AI agent for each student which has been told (in its system prompt) to contrast the student's approach with any particular theory and point out how and why the student's approach is to be modified to meet the accepted solution.

While the 5 stages of Deweyan cycle are already there in this approach; we also want to have *instrumentality* of concepts for the less curious, grade oriented students. To do this, the teacher can do a depth-first traversal of the concepts needed to solve a hard last-year question (a goal which many students would want to accomplish). This will go as follows:

- The teacher presents the question to the class and asks the class about what things seem unfamiliar in the question (asking for where to go next) and notes it down.
  Then he starts explaining the first concept in the list he noted down. Once again, he should ask the students what feels unfamiliar. This keeps on going till the students don't find anything unfamiliar, that is, a *primitive concept* has been reached.
  - The teacher then pops this concept from his to-teach list/stack and starts teaching the last thing that the students asked questions about. If yet another question arises, the teacher should put that new thing on the to-teach stack.
  - After every "major" (depends on context) concept is completely finished, the teacher might put assignments or quizzes *only* on that concept.
  - Eventually every pre-requisite for the question is finished, and so the teacher can ask the students to solve it and later *show* the kids how to solve it.

  This, top-down; recursive approach to teaching rather than the traditional bottom-up approach helps give the students incentive to study the many seemingly un-related primitive concepts that are needed for the derived concepts and ultimately the last-year question. Instrumentality of the current concept is present at any given moment in this manner. A lot of books are in-fact written in this manner; and experts (at least software engineers) often work in this top-down manner.

▼ Why Minimal Guidance During Instruction Does Not Work

- Cognitive Load Theory (CLT) says that learning is changes in the Long Term Memory (LTM). In particular, the scheme is stored in the LTM, not in the STM.
- Since learning is changes in the LTM, reducing the load on Working Memory (WM) that is required to change the scheme correctly is efficient learning.
- CLT says the unguided approaches such as learning by doing (constructionism), questioning (inquiry) and reflection are only possible after sufficient schema has been built up that enables a learner to link the experience to other experiences well (already explained by ZPD, but they didn't read this apparently). Otherwise most of the work done in the STM will be useless
- The paper accuses the unguided approaches of not having scientific basis, since they are grounded in pedagogical ideology and not cog. sci.
- The paper uses lab based studies as its grounding. Thus, it doesn't have ecological validity like pragmatism. But at least let's hear them out.
- Expertise Reversal Effect (ERE) : when the learners become experts, too much guidance actually hurts because it conflicts with their inner schemes sometimes (classic constructivism)
- The paper acknowledges ERE but says that for novices this won't happen and that there is no such thing as "too much guidance"
- The paper says that the top-down inquiry based approaches are "unguided" and go even further to say that in such approaches, the teacher is basically absent.

This is factually false. Papert explicitly stated that the teacher is burdened with creating circumstances that invoke construction based inquiry of the thing *we* want.

- Both Vigotsky and this paper agree that kids can't directly copy experts without pre-existing schema and thus, a purely self-sustained growth is practically impossible.
  Vigotsky's (mature) solution to this is scaffolding and ZPD.
  The paper's (immature) solution to this is instructionism and modeling.
- According the CLT (this paper) the decision making done by the learner when to seek optional scaffolds are "extraneous cognitive load" whereas according to Dewey and Freiere, or even the activity theory / cultural pathway stuff, this is *agency* and is extremely important. It's important for the kid to appreciate what the thing that the scaffold provides and that won't happen if someone barges in and gives them the info; that will make them irritated and un-confident at best. The knowledge *must* be constructed by the kid or he will be under-confident and dis-identify with the actual practice. They are not considering these things.
- The paper falsely accuses constructivists believing that all guidance interferes with the natural and effecient process by which learners inquire and learn and thus build style.
  This isn't actually what the constructivists believe. They says that guidens puts *constraints* on the solution space of the schemas possible. This is both good and bad, just as being given a hint for a puzzle is both good (ZPD) and bad (avoiding another possible solution/ style).
  - There are two assumptions which they pollute with their interpretations:
    - Dewey : Students should solve authentic problems (reflective experience) for maximal learning.
      They interpret it as "student's should solve problems *as seen in real life*, meaning, with the same level of complexity. So they are basically using the "no one becomes an engineer in one day" argument to "falsify" this "assumption".
    - Constructionism : Students should solve problems in similar context as they are found in real life.

      They interpret "smiliar" as "identical" and then use their "lack of concepts" argument again, completely ignoring the fact that Papert was saying "similar but easier (sequencing)" and not identical.
  - The good stuff that the paper states
    - Constructivism is right in saying that learners construct knowledge on their own and cannot be programmed.
      But that doesn't mean learners should have to *discover* knowledge themselves (basically saying that discovery learning is impractical, which it is)
    - While it is true that the final practice is what we are trying to teach, it is not necessary that we should have the pedagogy identical to the enterprise (talking about sequencing and ZPD here)

▼ Preparation Assignment Week 10

Is Instructionism always bad ?

Before I state my stance on Sweller's claims about Minimal Guidance, I believe I should first answer this question since a lot of what Sweller says seems to support instructionism and the theories we have read so far are all anti-instruction.

I believe instructionism is often necessary for high risk feilds such as farming, fishing, aircraft pilots, drivers, etc. A lot of communities of practice are also based on instructionism. For example, traditional dance communities are almost entirely instructive for the newcomers. Similarly, any martial arts community is instructive for the most part. These communities work well and the newcomers don't dis-identify with the practice.

While Dewey says that perplexity is a necessary component for experience, I believe it is "desire" or "incentive" that is the root cause of any experience. Most kids who learn how to play a piano piece really well or how to speed cube or even how to solve arithmetic really fast are not really perplexed as to how to do it. Instead they do it because "it's cool". While Dewey will not call it an experience since a speed cuber is not really going through any inquiry while his simple practice, it is almost certainly a *happy* and a *cognitive* change.

I am currently enrolled in a course for learning Blender. The concepts are taught to us in a bottom-up approach. We are first told how to do something, are made to walk through an example and then given assignments which we can do using the tools learnt in various ways. We are not given the assignment first and then told to figure out how to do it, thereby learning concepts. The reason for this is that everyone already has an incentive to study the procedures and tricks and don't need more incentive to do it. Unguided assignments would force to rely on what we know already, and while it may strengthen those concepts, they won't make us seek the optimal strategy. The problem that worked examples and instructions solve here is the expansive search space and the extra cognitive load associated with

searching things by ourselves, choosing the best one, picking sources to read from, figuring out exceptions and easter eggs through unnecessary trial and error and overall struggling and feeling frustruated much more than we need to. From this list of bad effects that I gave, most happened with me when I tried learning it on my own some two years ago.

The message is clear; unguided learning causes extra (cognitive) load and often frustration, potentially causing dis-identification and believing the task is “too hard”.

Coming back to my point about incentives; although when learning in a bottom-up approach (schema first, application later) the acquisition of schema is not as rewarding as compared to the Deweyan top-down approach and might even remove some sense of agency from the learner since he is now forced to forced to do things that are meaningless (abstract) in the short term (and will become very useful later), it is undoubtedly more efficient and provides a sense of security (structure) to the learner.

Guided discovery

While *unguided* discovery is obviously harder and tends to fail a lot, guided discovery does not necessarily lead to failure. I experienced guided discovery routinely when I was practicing for JEE. The worksheets I solved very well sequenced and I could (re)discover most of the 12th grade mathematics and physics all on my own simply by working through the questions. JEE problems are also known for employing problematisation a lot to “trip off” students with un-refined concepts. This unintentionally also serves the students since the practice questions are also similarly problematised and lead to perturbations, and perplexity and usually ends with either more steps added to an existing method (constructed by the students) similar to adding more code to an algorithm for edge cases (bricolage), refinement of concepts (similar to accomodation), or even picking up or rediscovering a new method or generalisation (in the ZPD).

This is a good point where I should point out that the studies cited by Sweller were all comparing unguided discovery of a concept much outside the ZPD with either worked examples or instruction.

One study in particular gave a physical setup to the two (guided and unguided) group and asked each to optimise for a particular effect based on some variables in the setup. The unguided group was unable to figure out how to model completely, but made some slight progress. The guided group on the other hand was given the model (the result of reflective-abstraction done by the *scientists* who created the model) and were successful in the task. The same two groups when asked to do the same thing again for a slightly different setup which *also* fit the model showed the same result again. The researchers concluded that “unguided dicovery doesn’t work” while the real conclusion should’ve been “unguided discovery doesn’t work when the sequencing is trash and the task is miles away from the ZPD and the guided group is spoon-fed the surface level practical schema for 100% efficiency in a narrow set of tasks without any concern for integration of this schema into their existing schemes” .

While Sweller uses the CLT argument for justifying this, saying that the unguided learners are overwhelmed due to the complexity of task, he is actually making a conclusion in similar fashion to a behaviourist. For him, increase in efficiency seems to imply increase in learning. He believes as long as the schema gets printed on the LTM somewhere, it counts as knowledge. But from what we know from the experiment done on the street-dweller kids who were asked questions involving mathematical concepts but framed as money management problems, this isn’t the case. The context that this schema was learnt in matters. We know that most people maintain micro-identities which can be thought of as partitions in the full schema. For the kids who are “guided”, they form a big split in their schemes approximately dividing it as “real life” and “schools and textbooks”. We don’t want this to happen. If the students are unable to incorporate the schemas achquired from instruction into their *working* , *real life* parts of schemes, then they will never believe in what they learn and will never have enough confidence to apply any of it to a new *kind* of problem.

Another issue is that just giving the end-to-end procedures as black boxes doesn’t allow the student to learn the subroutines which are often more important and lead to easier transfer. Guided discovery on the other hand, forces the kids to discover them.

Worked example effect

Sweller mainly prescribes the modelling part of the cognitive apprenticeship and calls its effect as the “worked example effect”. He warns us of the “Expert Reversal Effect” as well, which is when guidance to experts becomes hindrance rather than help; but that is very different from the critiques of instruction that constructivists give, which are mainly the issues of agency and improper integration of schema with the working schema (I will be using this self made term a lot since I have not other way to describe my thoughts).

As is evident from my account (Blender class), the worked example effect does decrease uncertainity, frustration and cognitive load on the WM.

Unfortunately, Sweller and company seem to be ignoring factors such as motivation arising *from* the education and not the other way around, and the fact that only worked examples create robots of students who are unable of critical thinking and praxis, and also things such as confidence in solving new problems.

While the worked example method works well for simple problems that require no decision making or real struggle, it won't ever work in the messy situations that experts actually go through.

Their paper states that the enterprise and the pedagogy for that enterprise shouldn't be expected to be identical and thus refute claims such as lack of validity of schooling knowledge in real world while not giving any account of *only* worked examples succeeding in closing the gap between theory and practice in the intellectually demanding fields such as STEM.

Structuring

Reiser describes 2 modes of scaffolding both of which are used a lot in university level textbook problems. Reiser is mainly focused on how to do these both using software (similar to Papert and Logo). Structuring is when the an outline of the process is laid out by the software itself which allows lower cognitive load at any given time on the student. Helper tools such as calculators and spell checkers also fall into the structuring category since they abstract away the tiny details that would've otherwise caused unnecessary cognitive load. Structuring reduces the solution search space by limiting attention only to the high level objects, and planning on how to use these high level objects/algorithms to do the task. He also puts "focus improving" scaffolds, which dis-allow wandering off into tangents and trying too many things and "monitoring" scaffolds such as timers, test score charts, per-problem timer, accuracy charts, etc. into the structuring category.

He also mentions structuring social interaction through the tool so that the students can understand the proper language for the practice. This is supported by Vigotsky's theory.

The best example of such structuring is StackExchange. The activity system has very explicit rules which are ruthlessly implemented. While this does deter many away from participating, it leads to incorporation of the rules as signs eventually.

By structuring, we not only limit the solution search space but also provide new schemas to the learner that they are to incorporate and use as the sequencing of the tasks proceeds.

As opposed to work examples, structuring is more general and doesn't necessarily spoon feed the learner allowing for some style and critical thinking to still take place.

Problematisation

While a common interpretation of it is "tasks that force you to reflect on what you did" it should not be taken too far since applying actual force; say grading the reflection for example just takes away the point of reflection. In that case, the learner will only use textbook (or worse, AI) reflections and not really learn. Instead, I would interpret problematisation as a "schema refining pertrubation". It's aim is to force critical thinking and refinement of the schema acquired and assimilated through structured problems or worked examples. It does match with the reflection/update part of the Deweyan cycle, but need not be an externalised, systematically written reflection. Instead it is more of an event or problem that leads to comparison and conflict between two (or more) methods or concepts learnt so far, leading to refinement of both. This refinement and accomodation is conscious and needs critical thinking. With enough of a perplexity, this may even lead to inquiry (asking peers, online community, etc.). I saw a lot of this happen during my time at the PACE coaching institute. As I already mentioned, JEE problems are known to be "tricky" (problematised) and this would often cause doubts that would be resolved with the help of the community and teachers.

My stance

My views align the best with Reiser's views. Moreover I also beleive that we already possess enough technology to build self-sustained learning environments as the ones he cited.

▼ Designing for Productive Failure

▼ RSM

A Representation and Solution Method is a particular strategy/concept/sign applied by a problem solver to attempt a problem (for example analytic modeling, equations, algorithms, hit-and-trial, intuition, etc.).
This thing is outwardly visible to others most often and is usually built on top of understandable language and can thus be articulated easily.

▼ RSM diversity

Usually a person will try multiple RSMs for a problem until they come to an impasse (no idea coming to mind). This number (averaged over problems) is called the RSM diversity.

▼ Mechanism A : Differentiation of concepts

When a solver tries multiple RSMs, he notices different pros and cons of the RSMs as well as what the RSM can solve and cannot, and where it is applicable and where not and why.

Thus the problem solver differentiates between the concepts involved in the RSMs he attempts as solutions.

▼ Mechanism B : Critical features

Different RSMs have different features.

▼ For example

- hit-and-trial is high risk and high reward
- brute force is good when numbers are small and bad when large
- writing equations help for complicated scenarios and generalisation but not quick-and-dirty math (like that done on the street)
- ratios like speed are used for comparison and need to be converted back to distances and time before they can be added

When a problem requires a particular set of features in the RSM used, it is called a critical feature.

Critical features can be found out via group discussion, articulation-and-reflection, comparison with a model solution, etc.

▼ Mechanism C : Explanation and Elaboration

Group discussions allows one to see and compare RSMs employed by others too, leading to extra RSM diversity.

According to constructivism, perturbations are necessary for accomodation (qualitative changes (not just addition of concepts) in the scheme).

Group discussions is an incredibly fast way to allow students to get more perturbations *and more RSM diversity* .

It’s also easier since the students will challenge each other’s solutions, leading to articulation, reflection, and eventually correction of misconceptions.

▼ Mechanism D : Assembly into targeted concepts

When a canonical (”correct”) solution is given, the students can compare it to their own solutions, modify their solutions iteratively, and eventually see all the critical features.

It should be noted that this process happens in a very contructionist way, (through bricolage) and leads mainly to the modification, combination, and extension of the original RSMs, and not acquisition of a completely new, unrelated (in the eyes of the students) RSM that lies very much outside their ZPD. Thus, the sense of ownership and style is still preserved.

▼ Phase I : Generation and Exploration

This is where mechanisms A,B,C take place. The students are given problems to solve, and a class full of other students doing the same thing. This leads to high RSM density, differentiation of prior concepts, and identification of critical features, effectively moving the ZPD so that the canonical RSM sits inside it.

▼ Phase II : Consolidation and Knowledge Assembly

This is when the students are either given the canonical solution or one of them figures out how to solve the problem in the canonical way through his experimentation.

This is where mechanism D takes place.

In this phase, it should be made explicit to the students that failure is okay and expected and the point of the exercise isn’t to just solve but to build understanding. Otherwise, an episode of failure may lead to dis-identification and lack of confidence and possibly alienation with the canonical RSM.

▼ Components of the design

- Activity : Choose correct complexity, (cognitive load), and respect the ZPD of the students. Ideally choose problems with multiple ways to solve.
- Participation Structures : Group the students based on their background (schemes they might have) so as to allow for productive discussion
- Social Surround : The learner's objective must be aligned with the activity's objective. The learner needs to be told that it's okay to fail. Moreover, we also want the learner to have a sense of ownership for his ideas.

▼ Preparation Assignment Week 12

Dewey and Productive Failure

The activity (the core layer of the PF design given in Kapur's paper) is the problem solving done by the students in the solution generation and exploration phase and the assembly into a final solution based on the critical concepts found in the canonical solution given in the assembly phase .

This problem solving happens before any instruction and is mostly unguided. The students are expected to fail in the first phase.

The first phase fits the definition of a reflective experience very well.

- The perplexity stage in the Deweyan cycle is the (yet unsolved) question itself.
- The hypothesis stage in the Deweyan cycle is the adoption of a RSM based on the critical features needed (that the target concepts in the canonical RSM provide).
- The experimentation and data collection stage, is the usage of the chosen RSM to attempt solving the problem and identifying the pain points. This is improved by by explanation and elaboration of different RSMs employed by others in a collaborative setting.
- The reflection/update stage is figuring out new critical features needed in case the RSM failed to solve the question.

In phase II (assembly into target concepts) the same cycle is visible except the canonical solution is available. It should be noted that in phase II the students actually *modify* their approaches to incorporate components from the model solution rather than simply following the new method without any reflection. Thus, it's *their* methods that are just being transformed to align with the canonical solution. Basically, the knowledge is constructed (rather than instructed) through a process that mirrors the Deweyan cycle.

A point that Dewey's philosophy disagrees with the activity in PF is the nature of the problem. According to Dewey, the learner must have personal stake in the problem; otherwise it's someone else's problem they are solving. But PF on the other hand says the the problem's complexity/difficult must be neither too high (avoiding anxiety and overload) nor too low (avoiding boredom). PF believes that the students need not have a personal stake in the problems, but only be interested enough in them.

Here, I want to point out the reflective-abstraction level instrumentality of schemes (concepts). We (humans) are hard-wired to try out and refine our knowledge even if it's not necessary. This instrumentality is overlooked by focusing only on "personal stake" as a motivation for inquiry.

From a cultural learning pathway p.o.v., the (intrinsic) motivation of the students to solve the problem can be explained by their sense of competence and identity in relation to the domain.

Thus, I heavily side with the approach taken by the Productive Failure pedagogy.

My experience of Productive Failure

I was constantly going through productive failure for the 2 years of my JEE preparation.

While there was no social surround or participation structure, I still studied very much in line with the PF mechanism.

The knowledge required for the JEE examination can be broadly categorised as

- applied domain knowledge (ADK) : facts and methods that are actually used while solving problems.
- hidden domain knowledge (HDK) : facts and methods that are too cumbersome to apply during the exam but are used for proofs, understanding, etc.
- control strategies (CS) : methods that help in problem solving, such as drawing while reading (for physics problems), getting an analytical expression first before plugging numbers, converting values into SI units before

plugging into an expression, etc.

- heuristic strategies (HS) : probabilistic methods that help “solve” the problem by directly getting the answer rather than actually solving for it, such as elimination of options based on units or approximation, trying given options one by one by plugging into the question (for integration questions), etc.

An effective combination of all these skills is needed to score good. Moreover ADK, CS and HS are to be applied in unison, and not in isolation, giving rise to the combined skill, which I will just call “problem solving”.

During the last months of my preparation, I was given a massive corpus of questions to train on called the “Final Lap Modules”.

I can categorise my preparation components as

- practice sessions focused on improving problem solving (ADK,CS,HS)
- learning sessions focused on improving ADK and HDK

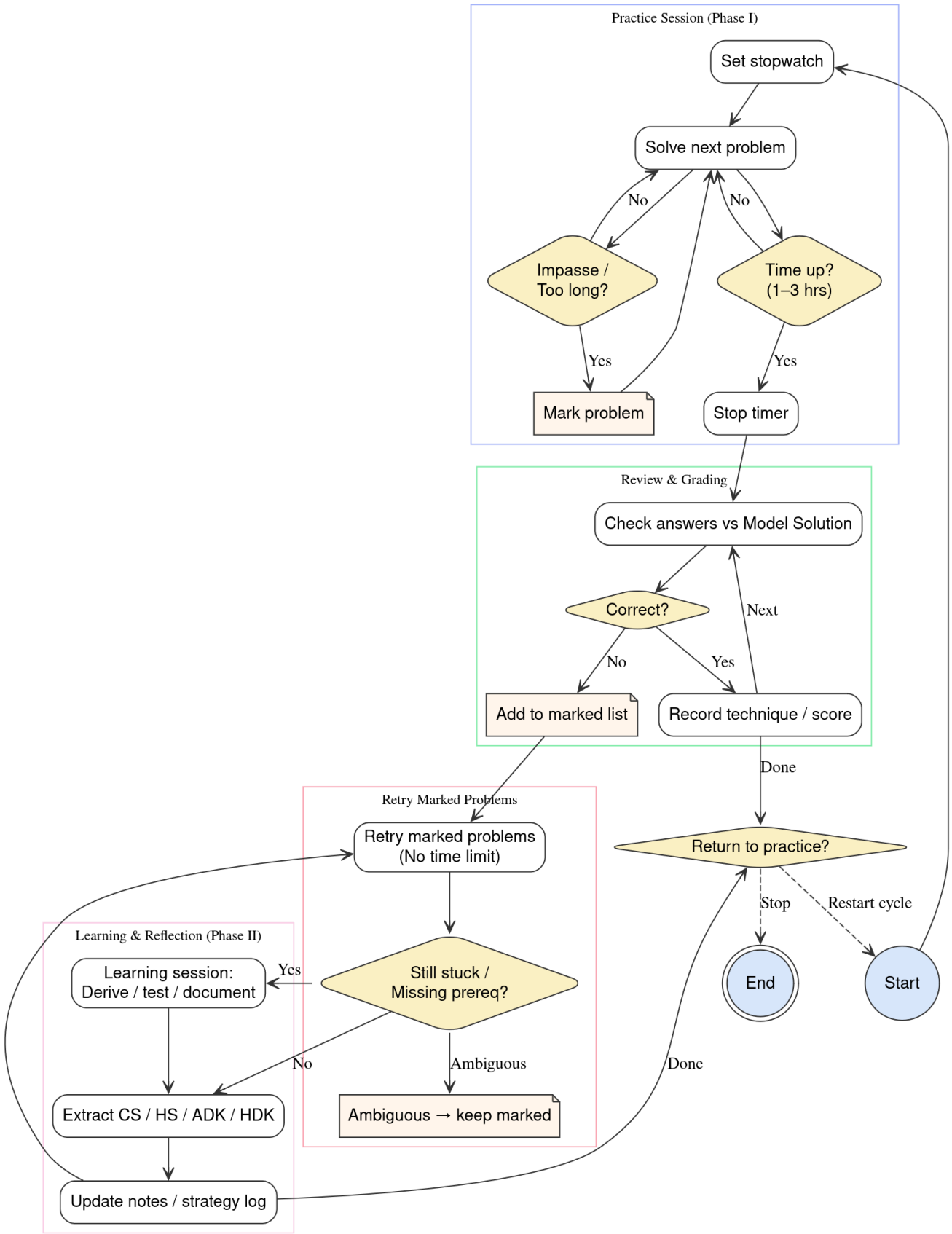
The practice sessions were the cause of the learning sessions which were (unknowingly) based on IBL. There was a lot of derivation, searching, trying, testing, formulation and documentation during the learning sessions. While these were the most interesting parts of my JEE preparation, since the topic is PF, I shouldn’t elaborate on this more.

The problem sessions were structured like this :

- Set a stop watch
- Start solving problems sequentially and solve as many as you can, only skipping a problem if it’s too long or you have reached an impasse. Mark such problems to review later.
- After 1 to 3 hours, stop the timer
- Check your answers (using the model solution) and thus grade yourself for this attempt as the net score over time taken.
- Try the incorrect problems again to see what could’ve gone wrong (silly mistakes, conceptual misunderstanding, etc.). If something is ambiguous or too hard to understand, mark that problem too.
- Try all the marked problems sequentially with a relaxed mind now and no time limit.
- For questions that you just didn’t have pre-requisite knowledge for, or are stuck on, do online searches.

The last two stages would lead to a learning session usually. Sometimes I would not have any marked problems and I would just reflect on the practice session, getting data for a new CS or HS that I might be experimenting with. This clearly resembles phase II (Consolidation and Assembly) in PF whereas the whole practice session is the phase I (Generation and Exploration).

This flowchart describes the process well :



JEE preparation process

▼ Collaborative Learning

▼ Different meanings of collaborative learning

- collaboration *for* academic stuff (quizzing each other, cross-checking assignments, etc.)
- collaboration in problem solving *leading* to learning as a side-effect
- collaboration leading to cultures, even if there is no intentions of the participants
- collaboration in a community leading to having that as a part of your history (cultural learning pathway ?)

▼ Ways to increase probability of interaction

- Set up initial conditions: symmetry in actions and status of participants, but slight asymmetry in knowledge.
- Giving explicit (over-specified) roles
- Forcing interaction rules in groupware to facilitate interaction
- Manually facilitating collaboration (think, a manager in a company)

- With many interaction cycles, the agents get to know more about each other, including similar or differing objectives. This eventually allows for offloading tasks to others *while* respecting their objectives.
- Cooperation is just the members dividing work either based on content or based on abstraction level and doing their parts. This is less interactive and roles are easy to maintain here.

- Collaboration is when there is a to-and-fro between the members and dynamic acting out of roles.
- Collaborative learning requires interactions to be real-time so that the internal schemas are exposed.
  For example, is CSCL, the chat is synchronous while emails are asynchronous.
- Collaboration often requires negotiability (ability to bend and learn new tools on the fly). This means being explicit as the task level (”what does equilibrium mean to you ?”) or a meta-communicative level (”what was the point again ?”,”Is that a claim or a comment?”)
- Collaborative learning *needs* misunderstandings for negotiation and learning (just like a mind needs pertrubation for accomodation)
- “Grounding” is the process of coming to a common ground (knowing what you want to tell is being understood as precisely that and vice-versa)
- Negotiation occurs after grounding. Negotiation is actually just negotiating for a new vocabulary/ideology that just works for all involved. It this can’t happen, we have a disagreement.

▼ Cognitive mechanisms involved while collaborating:

- Inductive abstraction : Abstract away your vocabulary and description of things so that others can understand it and construct their description of things that you tried to convey.
  When two functions (in programming) communicate, they don’t communicate the full state (abstraction) and they don’t communicate with the same variable names. All they need to do is understand the interface of the other function (induction) and then input their variables onto that interface.
  In humans, induction is actually ..inductive.
- Cognitive load : There is an overhead due to collaboration, just like there is a overhead due to managing an IO networking process (collaboration) with the actual task. But being connected to something else allows for distribution of load (even the trying out of RSM in PF is a load)
- Self explanation (and active recall) are ways to refine understanding. The question is, does co-explanation also help in refining the (absolute) understanding ? It gives a larger sample size for sure ..
- Conflict : Pertrubations but with the full team as the cognitive system.
- Internalisation (Vigotsky)
- Mutual modeling : Inductive abstraction is inductive and that induction arises after learning of the interface (vocab and beliefs). This “learning” is called “mutual modelling”.
- Appropriation is the conscious adoption of a scheme/tool/vocab used by the other person. While negotiation is only specific to that one interaction, appropriation is permanent. It’s conscious internalisation *after negotiation*.

▼ There are bi-directional links between different components of collaboration, namely

- situation (activity system scale)
- interaction
- processes (Vigotsky’s scale of analysis)
- effect (Cognitive scale..schemes, cognitive load, etc.)

▼ Preparation Assignment Week 13

During my compiler course project, I collaborated weekly with a programming language theorist. These meetings were not externally imposed but were self-organized, since I insisted on timely feedback and he was usually busy. The timing and structure of the meetings were therefore negotiated, not assigned.

There was a clear **asymmetry of roles and knowledge.** I, the programmer was responsible for implementing the compiler, he, the theorist was responsible for ensuring the design followed established principles. Our collaboration therefore did not have role symmetry, but it did have **mutual dependency**: I needed theoretical clarity to guide implementation, and he needed feedback from actual design attempts to refine what was possible within our constraints.

Each week, I would explain what I had implemented, what I wanted to add, and what the issues were. He would respond with what should or should not be implemented according to theory. These discussions often led to **conflicts**, since I aimed for a feature-rich, expressive language, while he aimed for textbook-quality design. These conflicts acted as perturbations at the team level, leading to negotiation and eventual refinement of our shared understanding of the task.

Grounding was crucial in every session: if either of us misunderstood the other's meaning, an entire week's work could be wasted. Therefore, much of our discussion focused on ensuring that what I said was understood as I meant it and vice versa. **Mutual modelling** naturally developed over time as we learned each other's vocabulary, reasoning styles, and constraints. This allowed us to align faster in later discussions.

Our interactions frequently involved **negotiation;** both about design choices and about the meaning of concepts. When I adopted some of his theoretical frameworks or reasoning styles as my own, that was an act of **appropriation**: a conscious internalisation of his scheme after negotiation. These appropriations gradually changed how I reasoned about design and language structure.

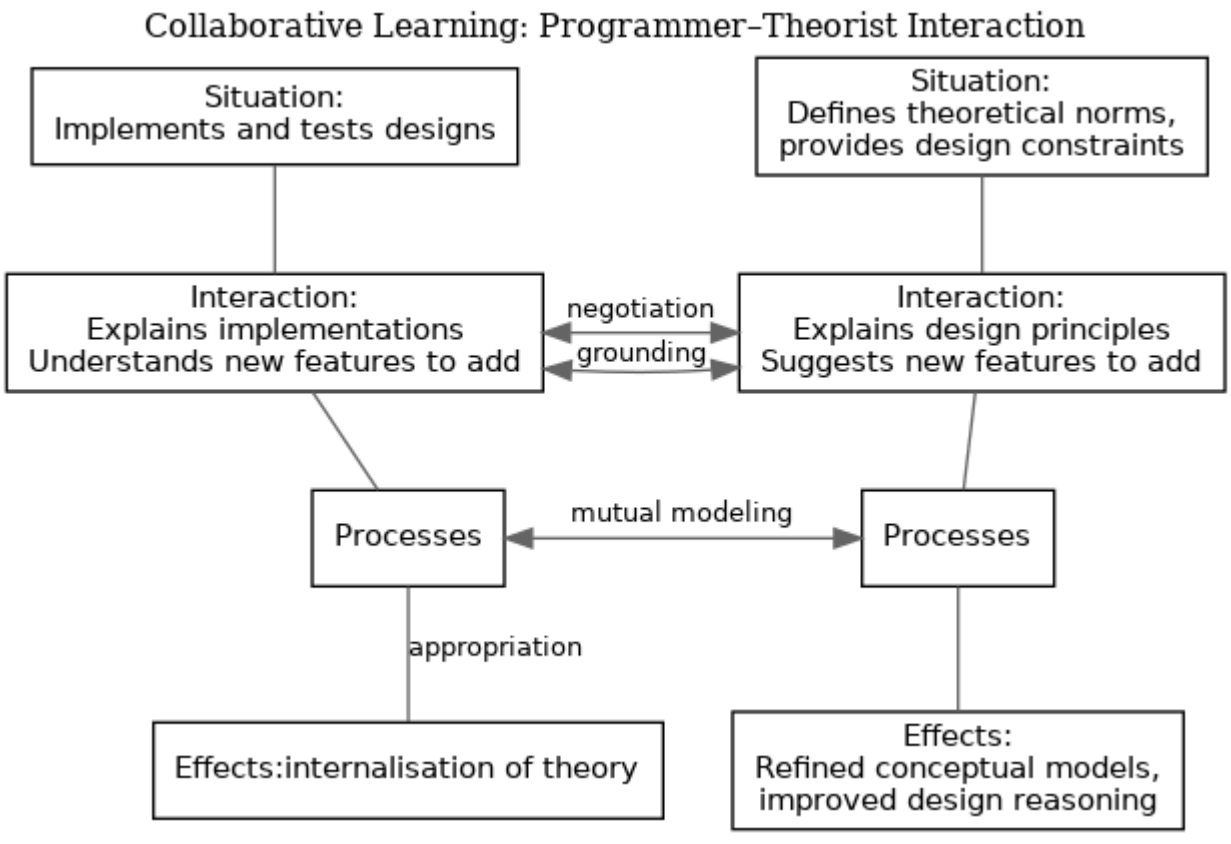
Several cognitive mechanisms were active:

- **Inductive abstraction:** I had to abstract my implementation vocabulary so that he could map it to his theoretical descriptions.
- **Cognitive load:** Collaboration added overhead, but it also allowed distribution of reasoning load between us.
- **Conflict:** Served as the main source of learning by exposing mismatched assumptions and forcing mutual adaptation.
- **Appropriation and internalisation:** I internalised several theoretical ideas after repeated cycles of negotiation and use.

At the **four analytical levels**:

- **Situation:** A self-organised collaboration in an academic setting, with clear asymmetry of roles but mutual dependence.
- **Interaction:** Weekly, synchronous discussions involving grounding, negotiation, and conflict resolution.
- **Processes:** Mutual modelling, inductive abstraction, and appropriation linking the interaction to learning.
- **Effects:** Internalisation of theoretical concepts, more efficient communication, and a refined shared conception of the compiler design.

The collaboration in this case was more of a mechanism aimed at a task and the learning was a side-product.



▼ IHLS Chapter 12

- A person may be motivated or engaged without being interested (doing it as a means to an end)
- Interest always leads to motivation and engagement
- Motivation is the individual's willingness to act (engage) in a given context.
- Engagement is done in three dimensions:
  - Cognitive (thinking about the actual problem)
  - Emotional (feeling in response to the activity) (this affects both cognitive and behavioural engagement)

    - Behavioural (participation in the activity system; even if no cognition involved)
- Interest is the relationship to the content of the activity.
- Components of interest
    - psychological (emotional) state during engagement; say, curiosity or flow.
    - Tendency to re-engage voluntarily (without motivation). This is a long term affinity.
- Motivation is volatile, but interest is not. Interest predicts long-term learning behaviours.
- phases of interest
    - triggered and situational interest
    - maintained and situational interest
    - emerging individual interest
    - well developed individual interest
- Interest emergence can be triggered by these motivations:
    - novelty
    - gaps
    - social context

    These are all properties of the activity itself and have nothing to do with the participant.
- Interest can be sustained by these motivations:
    - Identity formation (imagination)
    - a goal of mastery
- Timeline : Motivation → Engagement → Interest

▼ Preparation Assignment Week 14

Motivation, Engagement, Interest

Engagement refers to the effort put in an activity by a person in the cognitive, emotional, and behavioural/participation dimensions. For example, a student may engage behaviourally in the “study” activity by making clean notes, but not much cognitively or emotionally.

Interest, on a micro-level refers to the psychological state of the person while engaging and on a macro-level to his/her tendency to re-engage (without external motivation). Interest evolves in 4 phases: triggered situational interest, maintained situational interest, emerging individual interest, and well-developed individual interest. The first three map cleanly to the “trigger”, “immerse”, and “extend” phases in the micro-level interest loop in IDC (interest-driven creation).

Motivation is the *willingness* (not tendency) of a person to (re)engage in an activity, such as curiosity, imagining/identifying oneself in a role (self-concept), social context and support, and believing you have necessary competence (self-efficacy).

Motivation is often short-lived, whereas interest in more stable. Thus, interest is a better predictor of long term engagement.

How do these impact learning

Engagement in the domain is what leads to learning/competence/acquisition. Having a motivation to learn causes the learner to engage more that they naturally would. Micro-level interest is often seen in constructionist learning, where after immersing oneself in the activity, the activity itself becomes a motivation. Macro-level interest is related to identity and is sustained by self-efficacy and self-concept. An interested learner will learn with reduced friction and at a much faster rate compared to someone with mere motivation. To trigger interest, there must be novelty of the task, gaps in knowledge (challenge) and/or a supporting social context.

How can we design to interest students

The Interest Driven Creator (IDC) framework propose three loops.

The habit loop; cue-routine-reward/harmony; helps build a habit for the activity. I was experiencing this myself when I was preparing for the JEE and what I did in a day was entirely up-to me.

The activity itself is usually constructionist in nature and follows the creation loop; imitating-combining-staging/presenting. During my JEE preparation. I would imitate methods and then innovate by combining things I know to create better methods/explanations. I also had a YouTube channel dedicated to sharing of these creations.

Each part of the creation loop has the micro-level interest loop; triggering, immersing, extending. By extending the knowledge acquired during the sub-activity to other concepts (prior or future), the knowledge becomes meaningful for the student. For example, when I was studying projectile motion, I first derived the formulae using my knowledge of constant acceleration kinematics. Thus, there was some meaning added to the concepts learnt while studying kinematics.

IDC claims that their design will lead to not just better conceptual grasp (as most constructivist designs do) but also improved test scores. This, I don't believe to be true. I have several criticisms of their framework:

1. There is no assessment structure : They motivate their design using the exam-driven learning that happens in Asia. The reason for these exams is to assess the future competence of the student and select the most competent students, given the low supply of college/university seats. The authors of the IDC paper want to *replace* this structure with their creation-based framework. This wouldn't be an issue is it was as fair and scalable as examinations, but it is *not*. Evaluating creation is inherently subjective, and add to that the fact that the objective of the student is not actually to learn but to get selected, and we will have the students simply trying to match the rubric by using model "creations" to pass.
2. Not everything can be solved with constructionism: The creation cycle is at the core of their philosophy. While this works brilliantly for skills such as Programming and Mathematics, it isn't feasible for resource-heavy fields such as Physics and Chemistry and recall based fields such as history (at the level of the student).
3. They have not studied exam-driven learning well enough: Preparation for olympiads such as IOM and IOI are based mostly on interest. Each question is created to be challenging and to trigger, immerse, and extend interest. The same is true of JEE Advanced if its contents are taught well.

Rather than replacing the exam systems, if we were to increase the complexity and diversity of questions, so that the preparation induces interest, we would solve the interest/motivation crisis while retaining the fairness and efficiency of the assessment.

As for addressing the 21st century skills issue that they raise; that stems simply from wrong content being assessed and is not the main hurdle. We could replace theory based exams with practice based exams, that might involved programming questions, timed artifact creation (similar to "hackathons"), system design based questions, case studies, etc. and yet, we would be facing the same lack of interest if the questions on the assessment (and thus the kind practised) are "standard" questions based on mere recall.

This is the core of what I suggest as a solution. Other features that might be beneficial to add are:

1. Intentionally with-hold the test statistics of the full group from a learner and show the statistics that would lead to optimum engagement, self-efficacy and self-concept. Essentially, keep the learner/aspirant confident but not too relaxed.
2. Having excellent sequencing of problems is very effective. At least, it was for me.
3. A strong focus on the extend phase on the interest cycle helps make the concepts meaningful for the learner. For science and mathematics, this can be done via derivations.
4. A common platform for discussion (say, a WhatsApp group) helps give a supporting social context which mirrors the staging phase of the creation loop. I was fortunate to have such a platform.
5. Just like with test scores, the crowd on the common platform needs to be further divided into sub-groups so that self-concept can truly flourish since it is based on a perception that the person is uniquely competent in the field.

▼ Nov 18

▼ Reasons for natural start of collaboration

- The collaboration schema was learnt beforehand
- Some interaction cycles (in some other topic perhaps) was already done.
- The "collaboration culture" schema was there already. Both students believed in collaboration.
- Curiosity (about topic ?)

- synchronisity is not about whether the communication medium is sychronous but whether that tasks that people are working on are the same (even if in different roles). For example, cooperation is explicitely non-synchronous.
- Interactibility is a weaker condition where only cognitive processes of each other are affected due to interaction.
- If two people are trying to design something and adding features on top of current design, learning from seeing other's ideas, but not having any mis-understandings, doesn't that still lead to learning ? Articulation (self and co-explanation) and internalisation is still happening.
- Self-explanation and cueing is an internalised version of co-explanation and cueing other's cognition.
- Being stuck as a group is a "mis-understanding" ??? Yes.. because it leads to appropriation.
- Is appropriation like alignment in Du bois stance triangle ?
- Is it internalisation of task related concepts or the script itself ?
- Episodic memory is when a concept becomes tied to a situation. Collaboration *might* also have that as a mechanism.
-

▼ Creative Writing review

Learning Goals

It intends to make students conscious of the various genres of writing (prose, poetry, novels, etc.)

Design Features

- Writer's workshop (not in top down way)
- Instructor is co-writer (Arka Chattopadyay)
- No lectures
- Worked examples via readings (explained in lectures)
- Ungraded weekly assignment (only participation required)
- Normal Graded Exams
- Attendance
- Take home endsem (10 days) (project)
- Writing is seen as practice not just a skill

Learning Theories

- Constructionism
  - Sharable artifacts
  - learning by doing
- Cognitive apprenticeship
  - modelling
  - scaffolding
  - articulation in community
- CoP
  - identity formation
  - No belonging
  - Proff is old timer and some students are new-timers
  - Some students had already built and identity before the course

Implementation

- Bricolage
  Thinking what to write, reviewing it, changing elements (modifying slightly).

- Styles of writing were developed.
  For example, someone writing about a topic from a romantic angle and someone else writing it through a tragedy view.
- Shareable artifacts since everyone had to read what they wrote in assignments.
- Reflection opportunities were there but not in practice.
- Outline had to be articulated.
- Modelling : Reading,movies,classics,etc. were explained sometimes.
- One-on-one help was available in class
- Exploration absent
- Prof. brought in experts and even gave opportunities to connect with writers on national platforms.

Appropriation

- Since the course is creative writing *in practice* , it must necessarily have an applied component.
- The students need to know how to apply different things learnt and not just gain the schema.
-

▼ Course Reaction Survey

Merits

- The course content was relevant to real life
- There was no bias on any topics from the instructor’s side
- There was active discussion in lectures

Failures

- The assignments were not of the same complexity as the readings. Moreover the assignments required us to understand *everything* before writing.
- There were no sample solutions to the assignment. I had only chatGPT to refer to for correcting any mis-understandings, and that was not a good scenario.
- At some points it felt that we were just learning unnecessary vocabulary and observations, and not really trying to solve some question. For example, for the PF failure reading, we could’ve started with the data and studies, and then the theory.
- Some of the pair/group activities done in class felt forced and unnecessary.
- Collaboration wasn’t always present in the pair/group activities. I know I am an outlier for wishing for more structured pair activities, but in my opinion, it would’ve been better. The object of the student is mainly to finish the activity asap, and not collaborate. This is because we get very little time.
- It always felt like we were rushing to complete the syllabus. Perhaps starting the class with the activity rather than discussion is a better thing to do ? And are the activities even doing what they were meant to do ? There’s ample discussion either way.
- Differentiation between different concepts and preciseness was not focused on.
- In the sample critique, it was mentioned that the purpose of many of the theories taught at the start was not understood till later. I believe that with good sequencing of concepts and starting with questions rather than the theory itself, this can be solved.
- Often I found myself merely fitting a situation to the theories taught rather than actually *applying* the theories to decide what is a good design.

▼ Final Assignment

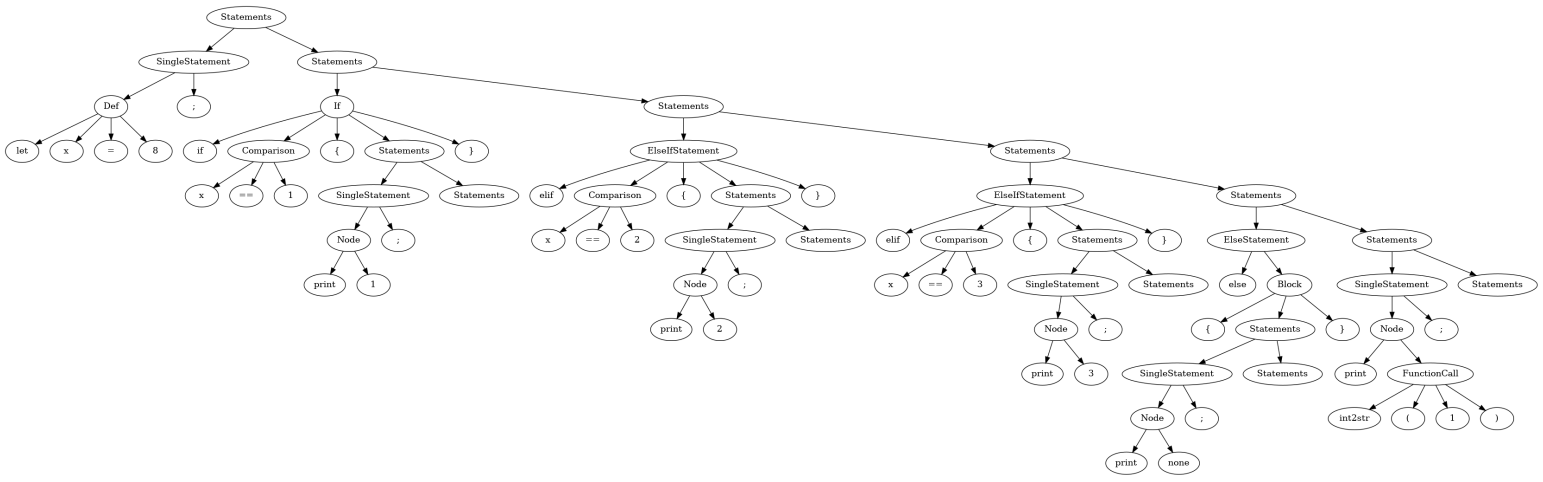
Learning Goals

Domain Background

Programming languages (PL) such as Python, C, Java, etc. are high-level languages (HLL).

```
Program

let x = 8;
if x ==1 {print 1;}
elif x==2 {print 2;}
elif x==3 {print 3;}
else {print 'none';}
print int2str(1);

Output

none
1
```

Code written in an HLL is either interpreted and executed immediately or is converted to a lower-level language consisting of instructions called bytecode. Optionally, it might be first converted to assembly code (e.g. MIPS) and then assembled into bytecode.




```
88  float_int_convert:
89      mtc1 $t0,$f1
90      cvt.w.s $f2,$f1
91      mfc1 $t5,$f2
92      sw $t5,0($s1)
93
94  end_int_convert:
95
96  # return
97  lw $t0,0($s0) # caller's base
98  lw $t1,0($s1) # return value
99  sw $t1,0($s0) # store return value at base
100 addi $t9,$zero,1 # assert that return happened
101 add $s1,$s0,$zero # restore stack pointer
102 add $s0,$zero,$t0 # restore base pointer
103 jr $ra # return
104 # End of return
105 label2: # end of function
```

```
1   00000011111000000100100000100000
2   00000011111000000000000000001000
3   00111100000000010110000000000000
4   00100011101100000000000000000000
5   00100011101101010000000000000100
6   00100010000100011111111111110000
7   00001100000100000000000000000000
8   00100001001010010000000000010000
9   00111100000000011000000000000000
10  00000001001010100100100000100101
11  10101110000010011111111111111000
12  00100000000110000000000000000001
13  00100000000110010000000000000001
14  00100010001100011111111111111100
15  00100000000010010000000000001000
16  10101110001010010000000000000000
17  10001110001010010000000000000000
```

A compiler is a program that converts code written in an HLL into bytecode. An interpreter is a program that executes HLL code without conversion to bytecode. A virtual machine executes the bytecode.

```
Program loaded successfully. 754 instructions (2.95 KB)
--- Starting Execution ---
none
1

--- Program exited via syscall 10 ---
--- Execution Ended ---
Total instructions executed: 376
VM Memory Freed.
```

Learning to create interpreters and compilers for PLs allows crafting custom PLs. While the set of possible PLs that can be interpreted is very large, the set of PLs that can be "sanely" compiled and can be used for general-purpose tasks and provide intuitive and necessary features is much smaller. Thus, PL design is an important task, and the theories used to design "good" PLs are collectively known as PL theory.

Course Context

This course, CS 327, is a CSE elective that falls into the computer systems basket and is aimed at students who plan to pursue careers in industry, where understanding compiler design and systems programming is increasingly valuable.

Professor's goals

> My primary motivation is to convey the beauty of the subject, get students interested, and then getting them to work on a large-ish codebase. This would be common to many of the systems courses if I teach them: compilers, os, networks, etc.. Problems that pop-up when developing in the large (with multiple people, spanning non-trivial time period) can only be really understood while writing large programs yourself. I think in Computing, DSAx, etc., there are no opportunities to do this. Main technical objectives are teaching lexing + parsing (a task that pops up almost everywhere in programming) and to expose students to some aspects of engineering with language design as a medium (types, mutability, first-class vs second-class).

From this, some goals can be identified:

- Convey the beauty of the subject and Get students interested : intrinsic motivation
- working with large codebases, exposure to tradeoffs in engineering, and skills (lexing,parsing) that "pop up everywhere" : competence in *real life contexts* (effective transfer)

Domain Knowledge:

While the goals articulated by the professor shed light on some domain knowledge. The actual things learnt and evaluated is more extensive. I am listing some major components.

- Understanding of basic PL theory
- Knowledge of interpreter and compiler design
- Mastery over intermediate stages such as lexing and parsing
- Understanding of VM and bytecode
- Practical ability to write code for interpreters and compilers in a HLL
- Ability to work with large code-bases

Control Strategies (soft skills):

- Teamwork and efficient distribution of work to accelerate product development
- Time management (over long time-frames)
- Managing the trade-off between usability/practicality and creative implementation (interest)

Students' Goals

In discussions with peers, I discovered diverse motivations for taking the course. Here I list a few:

- Industry relevance: Compilers and computer systems skills are in-demand in the software development market.
- Career incentives: Some classmates had secured internships at companies like Qualcomm with roles centered around compilers.
- Research interest: One teammate was conducting research in PL theory; this course was a natural continuation.
- Challenge and growth: Some students (including myself) had heard the course was challenging and saw taking it as central to becoming a strong computer engineer.

Overarching Career-Oriented Goal:

Given the goals of the students, a primary goal should be enabling students to actually take on compiler-related tasks in their future careers and possess the genuine competence needed to finish them successfully. To accomplish the

former, students need a sense of competence (confidence) and authentic interest in the domain. The same is visible in the professor's goals.

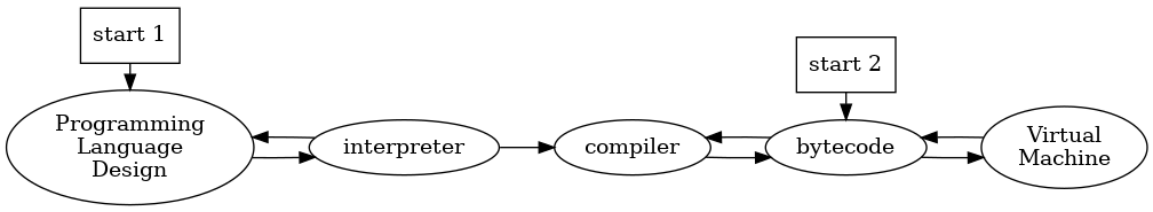
Thus, the core metrics that the students should improve on are:

- **Actual competence** : the ability to design and implement compilers in real-life contexts
- **Sense of competence** : confidence in ability to work on compilers
- **Genuine interest** : intrinsic motivation beyond course grades after seeing the "beauty of the subject"

Learning Design and Features

Structuring Through Sequential Implementation Pipeline

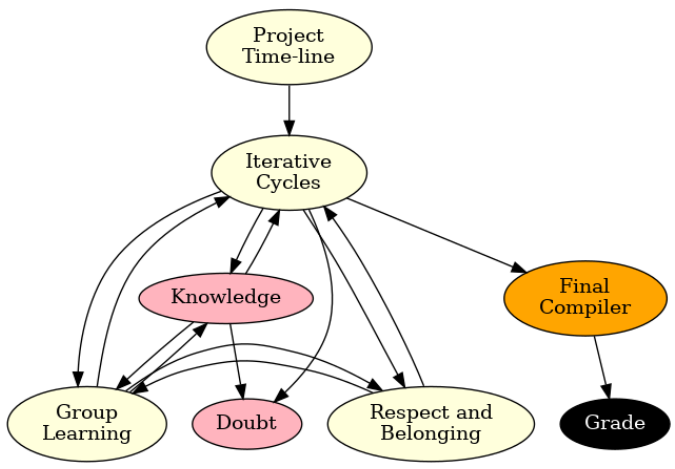
The course imposed a logical sequence on project work.



This structure prevented chaotic, simultaneous attempts at multiple implementation strategies. Instead, students focused on one stage at a time, building competence sequentially.

Assessment Weightage and Incentive Structure

The grading scheme heavily emphasized practical work over theoretical testing. Examinations received only 20% of the final grade, programming assignments received 30%, and the course project received the remainder (50%), fundamentally shaping student effort allocation.
The group project allowed for group learning, bricolage, and style development but by itself, didn't facilitate doubt resolution.

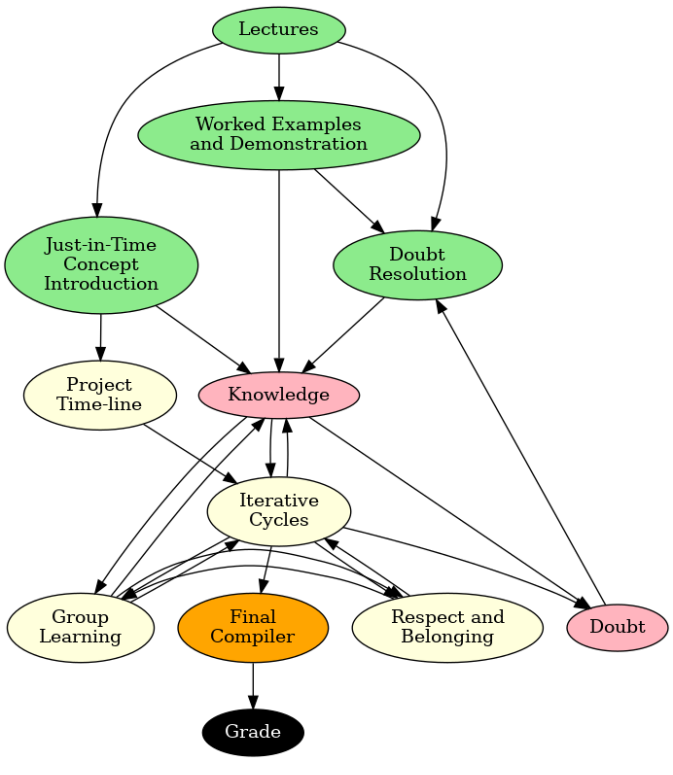


Learning Through Peer Examination

Student teams examined each other's PL designs and compiler approaches, using insights from peers to refine their own work. This created a some-what active community of practice where students were both learners and resources for one another.

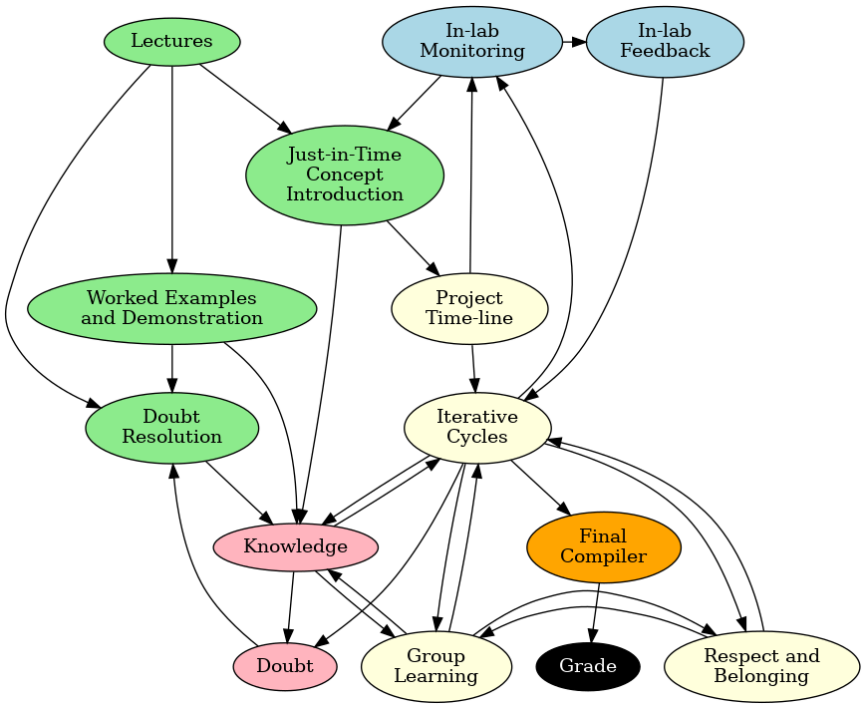
Lecture Structure

The lectures served both as doubt-resolution sessions where ongoing compiler work was discussed, and for introducing new concepts when they became useful to students' current work. For example, the cactus-stack data structure used for implementing function closures, was taught only after students had already begun implementing functions in their projects. This just-in-time scaffolding prevented cognitive overload while maintaining authentic need for the knowledge being taught. There was no grading based on lecture attendance; students were only expected to stay current with concepts introduced during lectures.



One-on-One Monitoring and Adaptive Feedback Through Lab Sessions

Lab sessions featured one-on-one discussions between students and TAs. These sessions were explicitly framed as monitoring scaffolds; forums for providing personalized guidance and feedback rather than for grading performance. This qualitative monitoring allowed the professor to provide adaptive feedback based on the teams' progress and challenges, similar to scrum meetings in industry or research lab sessions.

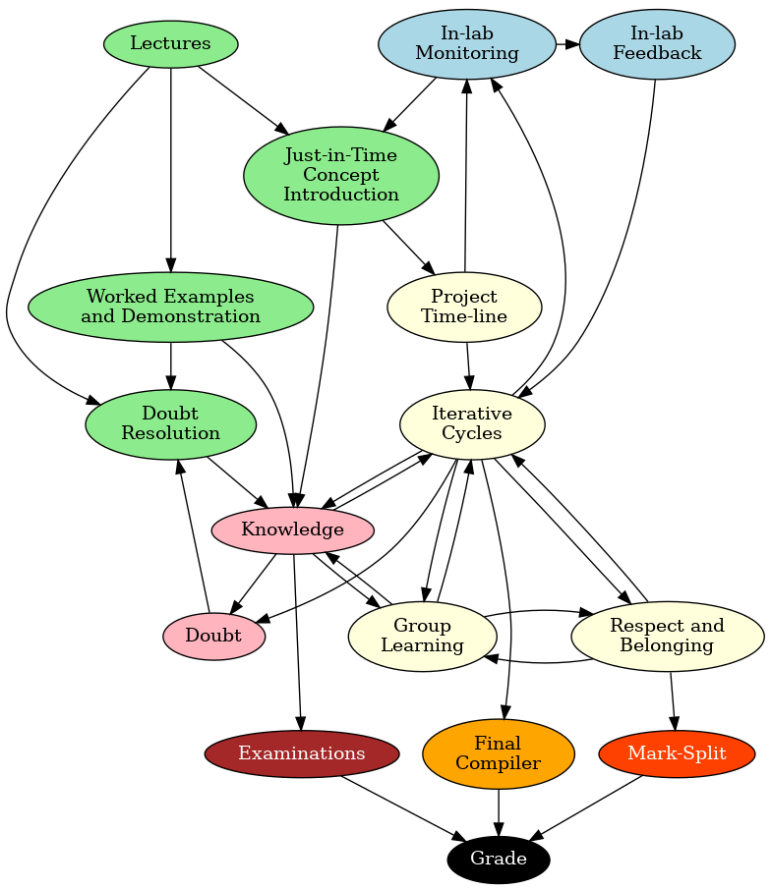


Theoretical Examination

Evaluation through the project suffers through plagiarism and excess reliance on AI. Thus, they are not sufficient to check the actual learning. Keeping a theoretical exam on a minimal set of topics that were (almost necessarily) learnt through the project solves this.

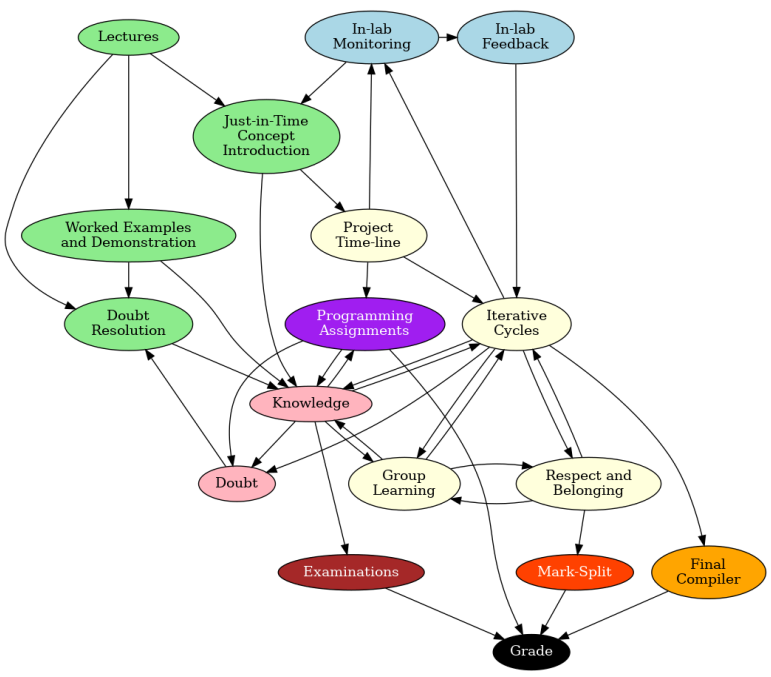
Fair Mark Distribution and Social Accountability

The course implemented a solution to the negative episodes (related to the practice of teamwork) caused due to the "freeloader problem" during evaluation, common in group projects: rather than assigning identical marks to all team members, the total marks for the project were given to the team, and team members negotiated and justified the distribution of marks to the professor. Critically, this rule was announced only at the end of the semester; had it been explicit from the start, it would have created unhealthy competition and discouraged team formation.



Programming assignments for Breadth of knowledge

While the project allows for deep internalisation and appreciation of a few useful concepts, it doesn't give a breadth of knowledge. This is solved by the programming assignments which are on a variety of topics. Moreover, because the assignments are much easier and mostly independent of the project, any weak student can gain new skills at any given stage in the project timeline and start contributing using these skills because of his/her comparative advantage. Thus, the "under-employed worker" problem is partially solved here.



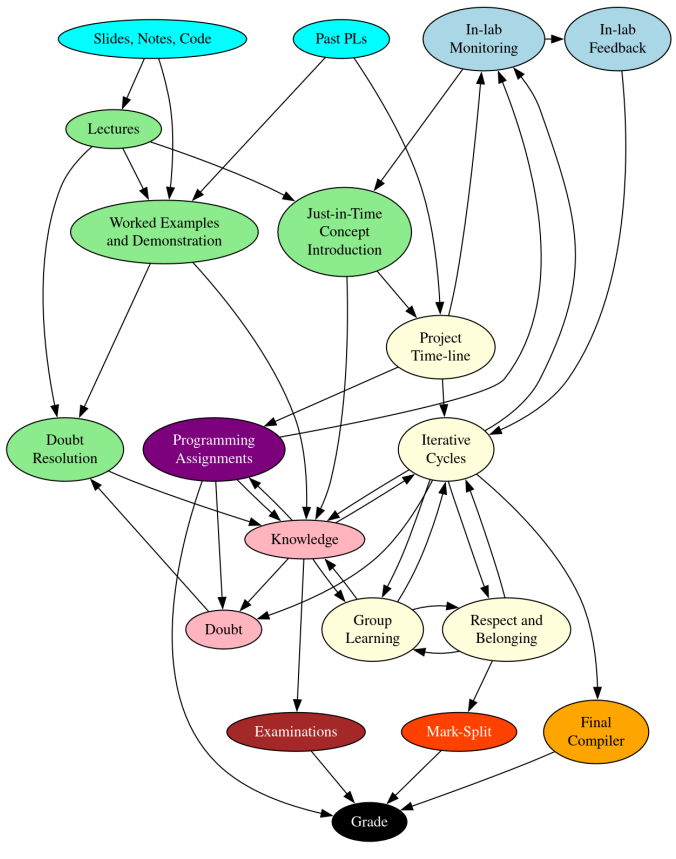
Other resources serving as worked examples

All course materials such as slides, notes, code, and relevant links were written on a page in a GitHub repository. At the top of the page, the professor explicitly stated:

> There is no textbook. Follow the lectures. Code a lot. Read necessary theory.

On the first day of the course, students were shown the PL designs built by teams in the previous year, providing concrete models of what successful final products could look like.

The code and pseudocode on the GitHub repository served as worked examples along with demonstrations done during lectures. This avoided a lot of confusion and anxiety that the student might have experienced if unguided.



Learning Theories or Pedagogical Models Used

Constructionism

Constructionism posits that learning occurs most powerfully when learners design, build, and refine tangible, meaningful artifacts in the world.

Evidence of constructionist learning can be seen from the course outcomes: by the end of the semester, I had learned nearly every concept required for the final written exam while referring to no written sources and attending only about half the lectures. The learning happened primarily through the project and assignments, not through instruction.

Learning followed a bricolage model, characteristic of authentic programming work. I would design or implement a feature, observe the restrictions and benefits that the feature provided after implementation, and based on that observation, either add another feature or improve the existing implementation.

Cognitive Apprenticeship

Cognitive apprenticeship, articulated by Collins, Brown, and Newman, describes how learners acquire complex skills through observation, guided practice, and progressively increased responsibility within an authentic domain. The course exhibits these components of cognitive apprenticeship:

- Global Before Local (Modeling at Scale) : On the first day, students were shown complete PL implementations from previous years.
- Modeling Through Worked Examples : Code and pseudocode for basic tasks
- Scaffolding
  - Timely adaptive help through monitoring in lab and concept introduction based on that
  - Structuring via the implementation pipeline (basic design, interpreter, bytecode, VM cycle)
- Articulation Through Lab Sessions

Cultural Learning Pathways

Beyond immediate course knowledge, both constructionist and apprenticeship approaches operate within the broader framework of cultural learning pathways. This perspective views the course not as an isolated event but as an episode within each student’s trajectory toward becoming a competent compiler engineer or systems programmer.

The course design respects and cultivates these diverse entry points. By producing genuine competence and belonging, not just grades, it creates positive episodes that shape long-term trajectories. My own experience illustrates this: the course’s success led me to ideate a new independent project (creating a parser for Oracle SQL).

Collaborative Learning

This view sees collaboration as not just cooperation (divison of labour) but a to-and-fro between the participants; leading to conflict, grounding (creation of a common ground), negotiation of views and ultimately internalisation of

knowledge.

By having the project be done in groups, the course tries to employ “collaboration” as a mechanism for learning.

Assessment of Learning Theory or Pedagogical Model

The theories employed in the design of this course have these benefits :

1. Constructionism supports development of style and a sense of ownership which leads to **idenity formation**.
2. Cognitive apprenticeship supports both **domain knowledge** and **meta-cognitive skills** through scaffolded, guided practice leading to practical competency **without overwhelming** the students as Sweller warns against.
3. The course creates a positive episode moves the students upwards in the cutural learning path by giving them a **sense of competence**. This “good episode” occurs mainly because of the constructionist approach which is known to be “fun” and often leaves the creator proud of his creation.

A comparative analysis of alternative pedagogical paradigms reveals that while many excel in some dimensions, only the combination of constructionism and cognitive apprenticeship produces all three critical outcomes for the over-arching career-oriented goals simultaneously (competence, sense of competence, and idenity formation) :

- Behaviorism and traditional instructionism produce procedural competence but lack motivation and ownership. Students can execute procedures but don’t develop identity or genuine interest.
- Cognitivism based instructionism supports mental model development but can remain abstract and disconnected from authentic practice, limiting transfer and motivation.
- Discovery learning can produce motivation but risks leaving critical conceptual gaps, especially in a domain as complex as compilers, thus not preparing the students for future.
- Problem-based learning engages students but often involves episodic problems rather than sustained artifact development, limiting the sense of ownership and long-term trajectory effects that come from building something persistent.
- Situated learning can overwhelm students with authentic complexity before they have sufficient schemas as Sweller warns about.
- Cognitive apprenticeship alone, while effective for competence development, doesn’t necessarily produce the sense of ownership and identity from having built something personally meaningful.

As for collaborative learning, although there was grouping, it didn’t necessarily lead to collaboration between *all* team-members, as was the case of my team. Even through I was lucky and reaped the benefits of timely guidance from a PL theory expert in the team, this is most likely a rare scenario. Thus the course failed to correctly apply the principles in collaborative learning. This is discussed in detail in the “design gaps” subsection.
Moreover, the reason for have the project be done in groups and the specific group size is unclear. It doesn’t lead to any significant advantages in my opinion.

Evaluation of Learning Design

Merits

Industry-level knowledge without abstraction overload:

The course operates on the implementable subset of PL and compiler theory, not the abstract formalism. This prevents cognitive overload from unnecessary theory. In contrast, the traditional format introduces full compiler theory, which many students find overwhelming and abstract.

Meaningfulness of knowledge throughout the learning process:

In this course, knowledge was learned because it was needed to solve authentic problems. I was able to learn nearly every concept required for the final exam purely through working on the project under guidance of the professor, TAs, and peers.
In contrast, students in the traditional lecture-based format often experience knowledge as abstract and disconnected from practice, reducing motivation.

Excellent sequencing and scaffolding and Appropriate complexity:

In many other CS courses, students receive vague project descriptions and must independently navigate literature review, design decisions, and implementation, or the projects have too high a complexity at the student's level. This often leads to overwhelm, shortcuts, plagiarism, and failure to develop confidence. This course provided careful scaffolding at each stage and avoids overload.

Alignment between assessment practices and domain practices:

Exams, assignments, and projects all relied on the same underlying domain knowledge, with minimal emphasis on exam-specific heuristics or strategies. Success in the project strongly predicted success on the exam, creating intrinsic motivation rather than forcing students to juggle disconnected skill sets.

Meaningful lab sessions:

The term "lab" is often used synonymously with "test", namely sessions where disconnected skills are graded. This course's labs functioned more like industry scrum meetings or research lab sessions: qualitative monitoring of progress for adaptive feedback rather than grading disconnected skills.

Fair evaluation of contribution:

The mark-splitting approach solved the free-loader problem during assessment while preserving motivation for teamwork. A team-mate who had done minimal work was upset when this rule was announced; I, having done roughly 70% of the work, was satisfied. Moreover my teammates were the ones who suggested I get a bigger fraction, signifying that they respected me. Thus the mark-split method evaluates not just the output of the team, but the perceived individual output.

Design Gaps

Meaningless Programming Assignments:

While the programming assignments introduced students to many different things, they were not directly related to the project and sometimes felt meaningless.

The Underemployed Worker Problem:

Students perceived as less competent by their team were not assigned challenging tasks and consequently contributed less, developed lower perceived competence, and potentially suffered negative learning episodes. Whereas students who did most of the work were overloaded and resented the ones who couldn't contribute.

This problem can be traced back to bad grounding and division of labour by the team at the start.

Teamwork is an important skill for the future activity systems that the students might enter and must also be *taught* or at-least be done by the course itself. This course fails to do both and might cause negative episodes in terms of the "teamwork" practice.

According to the collaborative learning theory, by structuring the assignment of roles, we can avoid vertical division of labour (which leads to assymetry in actions); unequal division of labour leading to higher or lower "status" of the student in the eyes of the team; and the extraneous load that comes from managing team dynamics.

Moreover, since we want collaboration and not mere cooperation, we need frequent shuffling of the roles.

We also need the collaboration component to happen synchronously (preferably face-to-face) so that the team members may be able to see externalisations of each other's schemas (say, while speaking), and not just the end product (a written paragraph or some simple statement with no reasoning visible) like we often see in asynchronous media such as WhatsApp or G-mail.

The Lack of Breadth Problem:

The constructionist approach is excellent for practical competency but not so much for gaining a wider range of knowledge.
The breadth of material covered was narrower than traditional courses. For example, recursive descent parsing was thoroughly taught since it was used a lot, but alternatives such as LR or RR parsing were never touched.

Redesign Features

Group based on forms filled out by students

While the multi-voicedness of the team leads to a variety of approaches and conflict, which leads to collaborative learning, it can also cause disagreements. So, have the students fill out a form with questions such as “what programming langauges are you comfortable with”, “have you built a parser before”, etc. and then let a computer program form groups, or let the students see the responses and decide amongst themselves.

Mutiple Implementations of Compiler Pipeline Stages

Each team-member should implement a different approach or component for each pipeline stage. For example, one member could implement a register-based bytecode and VM while another implements a stack-based VM.

Students can be graded with some weightage for their own implemention and some for the final compiler that the team will present using the mark-split method. This avoids the free-rider problem.

Have the students explain their own approach on odd numbered lab sessions and the approach of one other team-mate on the even numbered lab sessions. For the later, give 30 min of time to the team just to converse and prepare. The peer explanations can further be graded at the team level, forcing collaboration.

Benefits include:

- Easier distribution of workload resolving the under-employed worker problem via structuring of role assignment
- Exposure to a wider range of concepts and design trade-offs resolving the Lack of Breadth problem
- Opportunities for conflict, grounding and negotiation among team members.

Since this feature already solves a lot of the issues that the programming assignments were meant to solve, we can remove programming assignments, thereby avoiding the “meaningless assignments” problem entirely.

Extra Literature for Breadth

- Introduce “see also” sections covering ommited concepts.
- Keep these optional to preserve instrumentality. Students will be motivated to read these either as they face implementation choices requiring each method’s understanding or due to curiosity, and will undergo exploration.
- The final exam should only include concepts covered by approaches implemented by teams.
- Additionally, students can be asked to justify their chosen methods in a brief assignment (around 3% weightage), encouraging lightweight engagement with broader literature despite some initial resistance

New design at a glance