**Workplace Interaction Metrics Project**

By Stephen B. Gilbert

This project addresses the problem of how primitive we sometimes are when **we measure interactions** between people or people and technology. If we want to measure how good a team is at teamwork, we might count utterances by members and see if they’re equally distributed. That’s not really a good measure of team trust, but it’s easy to count utterances, so we do it. But is the measure predictive of team trust? Probably not. If we want to measure how much a person likes an app, we might count number of usages per day or number of taps per usage session. These measures also have problems and aren’t going to consistently predict app-liking, so we don’t reach our goal.

So in this project, we are seeking to **expand the state of the art** on how we measure interactions between people or between people and technology. These interactions could be on teams (remote or distributed) that could include both human members and/or agent or robot members. How would you objectively compare the interactions of team members on Team A with those of Team B?

You could also measure interactions between a single individual and a piece of technology, e.g., an astronaut and an automated maintenance repair agent or a pilot and an autopilot. If you choose the person and technology, though, the tech system needs to be complex enough that the project is interesting—the system needs to 1) have enough expertise that it knows things the person doesn’t know and 2) have some power over the person. Thus, you probably can’t propose using just a person and a mobile app, because mobile apps don’t typically have power over the person. (Feel free to pitch me something if you think otherwise.)

Because this course is focused on improving the cognitive engineering of workplace environments, mainly, the task that leads to the interactions should be **work-related**, i.e., not a sports team or a multiplayer game, unless the game is more work-related, like Artemis (a spaceship bridge simulation). Also, this project will make more sense if the people in your context **have a clear goal** for their task. It is likely that your metrics may help the people in your context do their work with fewer errors or more efficiency.

This is a group project, where groups might be 2-4 people, though 3 might be the best size. This project is worth 35% of your grade. This is meant to be an exciting project that you can integrate with projects you’re already working on, ideally. Please let me know if you have questions about tweaking the requirements in ways that will help motivate you or help make it more relevant to an existing project. Some of our results, as a class, might even be publishable when we’re finished.

To do this project, you’ll need to:

1. Define the work task and interactions that occur.
2. Define the roles in the interaction (how many people, agents, etc., and what does each do?)
3. Define metrics of interest
4. Give examples of applying those metrics to your task and the interactions.
5. Either evaluate whether the metrics you chose are predictive of what you want to measure, perhaps by correlating them with other measures, or describe in detail how you would evaluate your metrics if you had more time / resources.

You have two options in this project. You can:

1. **Build or Use a task simulation** (VR sim, multiplayer game if it’s work-related, etc.).
2. **Interview team members** **in the real world** doing the task.

For example, imagine someone says, “A good measure of how much team members trust each other is how often they ask for one another’s opinion about a work task.” You decide you agree, and you want to measure that. How might you measure that? You decide you will analyze speech signals and identify when sentence tone rises at the end (uptalk), because that would indicate question asking. So you:

1. Record spoken audio of each team member with separate microphones.
2. Segment audio into utterances based on pauses.
3. To detect question-asking, analyze audio signal for rising pitch at end of statements.
4. But research suggests that while uptalk indicates question-asking, it can be characteristic of geographic regional groups or low self-efficacy. So, you survey team members and filter your data based on these confounds.
5. Tag each utterance as question or non-question.
6. Tag each utterance with a Speaker and an Addressee. You could do this manually or insist that all your team members address each other and then identify Speakers and Addressees using a machine learning classifer, e.g., “Alice, what is your recommendation…” (Speaker = Sebastian based on voice; Addressee = Maria based on first word).
7. Count up number of questions asked overall within team.
8. Count up questions asked per team member to look for equal distribution.

***Consider Your Client***

If you think of this project as a consulting effort, you may choose a real-world workplace context that could benefit from your cognitive engineering skills, and you’ll have a client who will look forward to reading your final report with its recommendations.

That said, many real-world situations are too complex to solve in a class project. The time constraints may force you to do a more superficial analysis than you would like. And, this project risks focusing your client on pure numbers at the expense of the human side of their employees or users. Be sure to consider whether your approach increases that risk.

But, both the client and your instructor would be very interested to read about the assumptions or tradeoffs you choose to make during your analysis, as well as what you would do if you had more time and resources.

**Learning Objectives**

After completion of this project, you will be able to:

* Analyze a real-world workplace task in terms of cognitive load, decision-making, and triggers that lead to human error.
* Create metrics based on logs of real-world behaviors
* Evaluate whether the metrics are useful measures of the desired workplace construct

**Timeline**

Monday, Oct 5 by Midnight: One-page proposal on what you’re doing. Tell me:

1. Group name and who the group members are.
2. What’s the work task and what are the interactions? Between what?
3. Will you be building/using a simulation or interviewing real world teammates?
4. What’s the structure of your team? How many team members and what are their roles?
5. What metrics do you think you will use? (This may change during the project.)

Tuesday, Nov 17 & Thursday, Nov 19: Project Presentations during class. Add a link to slides in the Canvas page.

Monday, Nov 23 by Midnight: Upload deliverables: 1) PDF of Report, and 2) slides for presentation or link to video if they weren’t in the Canvas page already.

**Structure of Final Presentation & Report**

When you present, each group will have 15 minutes to present its work. Be sure each group member speaks during the presentation.

In the report, please include the following sections (single-spaced). You may wonder about page or length requirements. There are no specific requirements on length. Instead, consider that you don’t want to irritate your instructor by writing non-relevant material. Use the space you need to answer the questions below and communicate what you did.

1. ***Introduction***

**1(a)** Introduce the **task** and include pictures when possible. Include characteristics of the task such as:

* Is time-pressure present?
* Is this task done frequently?
* What subtasks are there?
* Are these subtasks interdependent? Does their sequence matter?
* How is successful performance evaluated?
* What are the consequences of high/low performance?
* Does financial compensation come into play?
* What motivates people to do this task?
* What are the interactions that you want to measure?
* How do these interaction relate to task performance?

Describe any contextual background needed to understand the task and its team members. E.g., is this in a hospital setting? A military setting? A retail setting?

**1(b)** Introduce the **players**. How many people/agents/robots are there? What are their roles? If there are agents/robots, is it tied to a particular human, or does it serve the whole team?

* Do these players know each other ahead of time (typically)?
* What skills do players bring to particular roles?
* How are individual players evaluated in terms of their own performance?
* What is the task allocation among players, and does it ever change?

**1(c)** If you have agents or robots, define them, e.g., in what ways does your agent qualify as an agent? What are the triggers to which it adapts? In what ways can it adapt? In particular, characterize your agent using the variables we’ve read about in class:

* Level of Automation
* Supervision method (how do you assign work to the agent?)
* Feedback method (how does the agent give feedback?)
* Actions of the agent (what is the agent capable of doing?)
* Triggers for the agent (what triggers the actions of the agent?)

Choose 3 metrics for evaluation. Ideally they are objectively measurable, not like a survey that asks the human, “How much did you like the agent, on a scale of 1 to 5?”

You might decide to measure the relationship between the human and technology or human and human, e.g., the team trust that exists between human teammates, etc. There are numerous things you could measure. You just have to specify. We’re interested in whether we can establish metrics for more complicated interactions.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Measure** | **Metric** | **Variable type (continuous, ordinal, categorical, etc.)** | **Data source (software logs, audio recordings, observations, etc.)** | **Collection frequency (every minute? Event-triggered? After activity? Etc.)** | **Threshold levels? Formulas?** | **Comments** |
| Commun-ication Balance | Ratio of max # of spoken utterances to min # of utterances on team per 10 minutes | Whole numbers / integers | Utterances recorded from push-to-talk microphone | Recorded through-out activity, then averaged | Count number of utterance starts for each team member in 10-minute period; divide max by min. | Higher (or lower) value does not equal better commun-ication; it depends on change over time. |

For each measure you develop, report at least the characteristics shown in the above table. You may add more.

There’s an example above that requires **machine learning for classification**. If you want to include a measure like that but don’t have the skills or time to implement a machine learning system, describe clearly what the classifier training set would look like and what the output of the classifier would look like. We can discuss this more in class.

1. ***Related Work***

Note any analysis done that is similar to yours, or instances where metrics similar to yours were applied. Are you extending the work of anyone else? Consider including topics from this course:

CTA & CWA, visual and auditory perception, mental models and knowledge representation, cognitive workload, attention, human error, adaptive automation, interfaces & controls, decision making, team dynamics & CSCW, data visualization. Cite at least 3 academic papers, either from this class or from outside sources.

1. ***Methods***

Explain what you did in enough detail that someone else could reproduce it and get the same results. Can you justify your methods with literature, e.g., “We used CWA per Bisantz…”?

Answer questions like:

* How did you generate your example interactions?
* What biases or assumptions did you make as you approached the problem? It is important to be aware of these and note them so that you can account for them later. One way of probing for this is asking yourself, "If someone else did this project, how would it turn out differently, and why?"

Either here or below, note what role each of your group members played on the project and you each contributed.

1. ***Results: Event Logs that Illustrate Poor and Excellent Example of Each Metric***

This section is where you put your **annotated event logs of interactions** and how you applied the metrics. If your interactions happen via text message, it might look like a text chat with comments on the side. If your interactions arise from using tools, it might look like an activity log in Word or Google Docs with comments on the side. The goal is to document the interactions and annotate them with comments about how a metric was measured.

For example, if you were measuring friendliness of a human-agent interaction, and you had the word “jerk” go by in a textual exchange, you’d show the event log and at that moment, you’d note that the friendliness metric would decrease by 1 unit, etc. I had difficulty finding visual examples of annotated event logs, but it might look like a movie script that’s been annotated with character movements, or a famous chess game that’s been annotated with the play-by-play, or it might look like a storyboard with comments. But it needs the event log needs to be detailed enough that it shows all the input data you use for your metrics. Some example ideas:

A close up of text on a white background

Description automatically generated

Figure : This script is annotated with some comments about intention and mood. You would annotate according to issues that affect your metrics. From https://radagastcasts.wordpress.com/2016/09/15/script-analysis/

A screenshot of a cell phone

Description automatically generated

Figure : This chess game activity is annotated with sportscaster-style dialog, but not really metrics like you'll do. From http://www.chessgames.com/perl/chessgame?gid=1008361

A close up of a newspaper

Description automatically generated

Figure : This information is not detailed enough, but it illustrates the storyboard approach of describing what happens over time. From https://www.nngroup.com/videos/ux-storyboard/

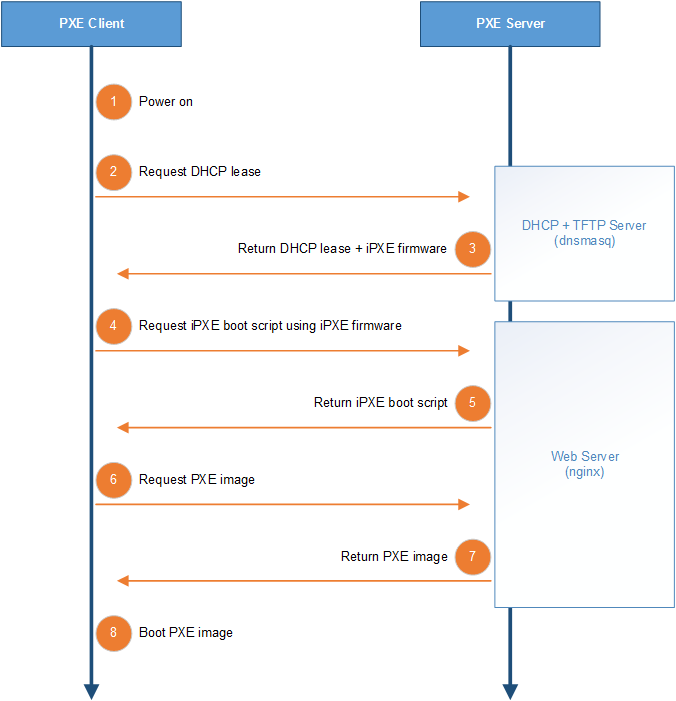


Figure : A network diagram in which time moves downward, with annotations along the way of what’s happening. From https://docs.01.org/clearlinux/latest/guides/network/ipxe-install.html

You need to generate detailed logs of interactions that will allow you to apply your metrics. You need to have **at least two separate event logs**; you may need more. For each metric, you need to show a time when the metric would show a low or poor example of the measure and a time when the metric would show a high or excellent example of the measure. You might have one event log that illustrates a poor example of Metric 1, a poor value of Metric 2, and a high example of Metric 3, and then another event log that shows a high example of Metrics 1 and 2, and then a third event log that shows a poor example of Metric 3.

1. ***Evaluation of Metrics, Limitations & Conclusions***

How reliable are your metrics? Would you use your metrics again? Can you validate them by exploring whether they are predictive of what you want to measure? If you can’t validate them for this project, how would you do that if you had plenty of time and resources? Are there other measures, e.g., performance, that would correlate with your measure?

Discuss generalization: Do you think your metrics would apply to other interaction contexts? Why or why not? What were the limitations of your approach? Is there something that prevented your work from being fully explored or generalizing?

Address whether your metrics would be useful for measuring the difference between experts and novices at your task. Why or why not? How would an expert score on your metric differently than a novice?

Address whether your metrics might de-humanize the work context that you analyzed. Could someone who is very numbers-focused use your metrics and end up ignoring important aspects of the workers’ humanity? Or could your metrics help with this situation?

State the contributions of each project team member to this project. Reflect on your overall approach. Was the task realistic, or representative of the interaction you had envisioned? If not, what would you have done differently as a project team?

Finally, what do you conclude from this project? Mention topics or research that contributed to your thinking and possibly research that you are extending with this work. How could your work (esp. if you had more time) impact other people? In what ways?

***Two Examples of Metrics***

**Alexa Example; Metric is *Familiarity***

In this example, I show a script of two people trying to work with Alexa to help them do their work. As a researcher, I want to evaluate how well Alexa knows the one human in this scenario, Alexa’s familiarity with him. My proposed metric for Familiarity is:

NOTE: This example script and metric definition are not well-done. They are hurriedly done in order to provide an example for the project. Please do better on your own. Also, I’m writing Alexa’s behavior based on a webpage about commands she responds to; they may not be accurate. If you use an available agent, please really test it.

Below, I’m highlighting the “facts about me” in yellow to illustrate how I’m counting for the metric.

|  |  |
| --- | --- |
| Carolyn: Ok, Baskar, we have 5 customers to visit with tomorrow. We should plan out who’s going to do what.  Baskar: Right you are. Ok, Alexa, what’s on my calendar for tomorrow?  Alexa: Tomorrow you have 5meetings. The first is Sulaiman Swimsuits at 8am. Would you like to hear more?  Baskar: No. [To Carolyn:] Ok, we’ll be Zooming with Qatar for that. Alexa, what’s the temperature in Doha tomorrow?  Alexa: Doha will have a high temperature of 110 degrees Fahrenheit tomorrow.  Carolyn: Wow, glad we’re Zooming! We’ll have to mention that in conversation.  Baskar: No kidding. How about I prep the pitch for them, and you do the financials that we’ll close with?  Carolyn: Ok. Alexa, How many Qatari Riyals to the US dollar?  Alexa: 1 US dollar is 3.64 Qatari Riyals.  Carolyn: Ok, I’m good.  Baskar: Alexa, what is the second meeting I have tomorrow?  Alexa: The second meeting you have tomorrow is Narasimhan sun dresses, at 10am.  Baskar: Alexa, read my Narasimhan to-do list.  Alexa: Your Narasimhan to-do list includes 4 items. First, send nature patterns. Second, get quote for double-printed linen. Third, Google celebrity outfits. Fourth, buy bacon.  Baskar: Bacon? Alexa, add buy bacon to my shopping list. Remove buy bacon from my Narasimhan list.  Alexa: I added buy bacon to your shopping list. I do not see buy bacon on your narrative list.  Baskar: Remove buy bacon from my Narasimhan to-do list.  Alexa: I removed buy bacon from your Narasimhan to-do list. | 2 facts re me, 1 reply  0 facts re me; 1 reply  0 facts re me; 1 reply  1 fact re me; 1 reply  1 fact re me; 1 reply  1 fact re me; 1 reply  0 facts re me; 1 reply |

***Discussion of the script and the familiarity metric:***

[I just make a couple brief points here, but you’ll want to grapple with your metric with slightly more length.]

In this session, Alexa had a familiarity for me of 5 facts / 7 replies = 0.71.

Is that a good score of familiarity? How would I know? Could I use this to compare Alexa with Siri or another agent?

Can my metric ever be > 1.0? Yes, if Alexa demonstrated many facts to me within single replies. So there’s no real cap on the metric, which is probably a problem. Maybe I need to have a normalized metric somehow. How might that work?

Did it make sense to apply my metric when Carolyn was speaking to Alexa, or should I have applied it only when Baskar was speaking, because it wouldn’t know things about Carolyn no matter what?

Would it be important to distinguish between Alexa knowing things about me that I tell her / give her access to vs. things that she knows about me due to surveillance of my browsing habits? The latter might feel more creepy than familiar. Can I factor that into the measure?

---- End of Alexa example

**Example with Surveillance Task.**

In this military exercise, there are two surveillance officers (“spotters”) stationed on top of a 3-story building in a small village. Each of them is responsible for monitoring any suspicious characters running around in their zone, the 180° half of the village that they face, either East or West. In particular, each spotter is responsible for “transferring” a threat to their partner. E.g. if you’re the East spotter and a guy with a gun is running from your zone towards your fellow spotter’s West zone on the southern side, you need to say, “Transferring at the south side.” Your partner spotter then needs to say, “Acknowledge” to indicate hearing you. Then, when your partner actually sees the threat with the gun enter the West zone, the West spotter should say, “Identified!” to indicate that the threat has been seen and is now being successfully tracked.

So, if we assume that there’s some active fighting or rebellion going on, multiple threats might switch zones and communications might look like this:

East spotter: 2 transferring at your north side.

West spotter: Acknowledge 2!

[pause]

West spotter: Identified 1 of 2.

[slight pause]

West spotter: Identified second of 2. Also, 1 transferring on your south side.

East spotter: Acknowledged.

[pause]

East spotter: Identified.

Let’s now say that the Army is thinking of replacing one of the human spotters with an agent, a robot with a swiveling camera called Bonnie. The Army wants metrics it can use to measure whether human-agent spotter teams are better or at least equal to the 2-human spotter teams. So really that means you need to figure out how to evaluate the spotter team, no matter whether its members are human or robot.

You decide on two metrics. Since the “right” communication pattern is a triple of “Transfer…Acknowledge…Identified,” you decide that:

As in, if the spotters missed a threat or forgot part of the triple, like the Acknowledge or Identify, Team Performance will be penalized. But you realize that if Team Performance is low, you won’t know if it’s because threats are getting entirely missed or you just have a team of spotters that isn’t good at completing all the commands. So, you thought it might be useful to measure team communication by looking at the number of times that spotters acknowledged each other. There might be teams, after all, that report on every threat, but they’re sloppy about it, not always acknowledging. So you also decide to measure:

You decide to record some spotter sessions. You use microphones on the spotters and have sufficient speech recognition to categorize utterances into Transfer, Acknowledge, and Identify, and you write software to log all those with timestamps. You end up with data that look like the following. These sessions last 10-15 minutes each, though, and you evaluate several spotter teams with several sessions each, so you have an enormous amount of data.

A close up of a map

Description automatically generated

To assess my metrics in this time span, I count up the numbers of each event across both spotters (the two sides, east and west). Totals are:

Threats on both sides: 13

Transfers on both sides: 12

Acknowledges on both sides: 9

Identifieds on both sides: 8

If we assume that whenever there is an Identified, there is also a Transfer and Acknowledge (is that a legitimate assumption?), AND

We assume that every threat that appeared swapped zones at some point (is that a good assumption?): then there are 8 triples and 13 threats, so performance is 8/13 = 0.62.

Instead, if we assume we can count triples by counting Identifieds (8) and instead assume that there were only as many zone swapping threats as there were transfers (12), then performance = 8/12 = 0.67.

The Spotter communication measure is 9/12 = 0.75.

For both of these metrics, are they measuring what I intended them to measure? Will they be good metrics for comparing two teams or comparing a human-human team with a human-agent team?

Are there metrics that would be better? Note that the current definition of these metrics must be calculated post hoc rather than in real-time. Is that ok? That means I can’t define team’s performance at any given moment. Is that ok?

Lastly, these are both team metrics, rather than metrics for individual team members. What if I wanted a metric for evaluating individual team members, e.g., so I could directly compare the Bonnie robot with a human team member? Maybe Bonnie is better than some humans but worse than others.

---- End of Surveillance Example