

P-I-T-M: Coma

WS 2004/05

Handout 6

30. November 2004

Handout6: Some remarks on the way we communicate

Ausgabetermin: 30. November 2004

Abstract

The handout is an ad-hoc document collecting observations and remarks (as far as we see it) concerning the current state of communication problems. The handout basically condense and makes explicit the stuff that has been discussed/presented during the meeting. So the document probably does not contain eternal truths, but just points that currently are of concern and which we should keep in mind so that the project(s) and discussions do not get out of hand. Compared to the distributed version, I leave out the *technical* points concerning the data model and concentrate on the “communication” part.

- **Request of “Stop the discussion”:** (by Tom)

- The discussion can indeed turn into a problem. As yesterday already indicated in the general email: underlying may be (amongst other things) a communication problem. One of the core messages I do support from what Tom (others too) remarked: we should come to an end. Furthermore, the discussions on the board start to get into an exchange of arguments for sake of the argument (“and here is yet another reason why I’m right and you are wrong ...”). We have to be careful here and come to a conclusion, in particular in cases where it does not matter what we decide (bitvector vs. integer, the arguments are on the table so we go ahead.)
- The second underlying message (“If we only had taken the carefully and painfully worked out good specification, then we would not had those problems”) I do not completely support (I pointed that out on the board). It’s true that the specs had been worked out well, but we are currently somehow deep in “Varchar(29)” (a typo) or “BigInt” vs. “Int” and all this kind of stuff, and also inconsistency remarks.

I would rather guess, that both of the two delivered specs would have drawn equal flak as well, once it came to the details or it came to realizing them really. We might have saved, I agree, a week or so, when just taking one of the specs and pour them into SQL (because the discussion would have started just one week earlier) but some discussion would have evolved.

The counter argument, “perhaps the previous specs had been some open points too, but at least *we* had to implement it” does not really hold in my eyes, because

we *all* have to implement it, not just each one his own spec. Life would indeed be easier if each group could implement whatever it chooses (spec 1, or spec 2, or some additions/subset), but that's not the idea.

- Related to the above concern (“general development, apart from technical details”) I further agree with Sandro (if that was his point): No unannounced critical changes which influences someone else. This holds even for *healthy* changes.¹

- And again related to the above: **the means of communication.**

- I can't speak for the rest of the audience, but it could be that this BB is not the fastest means of communication, in particular for *agreement*. This is not to propose email instead, but *sometimes* to come together and solve trivial technical things and that's it. What I mean is that for some issue the net time to follow long discussions, browsing, reading through quotes etc, is longer than the net time to sit together (or make an appointment with (me and/or Marcel and/or Gunnar), drop by and hammer it out. Also face-to-face sometimes makes compromise easier, and that's important as well.

- This was raised (by Gunnar) commenting on “agreements in the forum”. I agree with him in this point. Technically, since we (so far) said in the meeting, that the forum does *not* constitute an obligatory means of communication, there can't be *obligatory* agreements concerning topics of general interest on the forum.

A different thing would be that group *X* decides that everyone in this group *must* read the relevant group and thus the people are forced to use the board. In this point, I agree with *Gunnar*. *Scherbengestalt* mentioned: “I don't see what's going on” (that's perhaps an indication of internal communication problems) and “there are too many communication channels”. For the last point, that's difficult to change. The only thing that one can change is to reduce the *obligatory* channels (if someone likes to send someone else an SMS about the project, it's difficult to do something about).

- Here's the means of communication that we have:

- * **snv logs:** I do not recommend this as means of communication, unless we decide this, but I think it does not work, and we have enough other channels. Especially now, where somehow lot's of stuff seems to get checked in, we would be swamped by reading meaningless logs. As a matter of fact, many of the logs (by me and also by others) are plain empty.
- * **after-the-fact Readmes:** I'd recommend that Readmes are short and pretty *stable* (up-to the point of course, that things they describe change) but not used as a means of communication of changes. In this way, it would be more a **log-file**. Especially, it can turn out contraproductive, to change someone else's work and note it down in the README (I don't know whether we had instances of that already).
- * Also, in the medium run a bad means (or a sign of bad) communication: **“hand-versioned”** files like `file-x` checked in. At least the alternatives *must* as soon as possible been resolved.

¹Unless one really trusts each other.

In some sense, it's the polite way of avoiding the previous point (just overwrite or undo someone else's stuff and the other one has to swallow it or re-redo it). In the medium run, it's equally bad, because we will lose overview. For instance, currently we have 3 versions of the data-model in textual form lying around (two in the repos + one posted on the board accessible via some url) and 1 in SQL format. All versions are different. Version 2 seems to be the evolution of version 1 (so 1 is obsolete), one can guess that. Apart from that, however,² there creep in slight inconsistencies, for instance things that had been resolved in version 2, suddenly re-appear in version 3, which makes a proposal to solve one point in version 2, but *reintroduces* stuff that had been wrong in version 1.

The danger therefore is, that one loses overview (and we are talking here about only *one* (important) file. This means, it is (also) a problem of communication, because one forces the others to go through the code again, compare it, see new additions, see that old stuff has not been taken into account. Basically, it's the same situation as in the previous point, only more "polite" at the surface.

- * **Bugzilla** has not yet proved how helpful it will be, but I guess it is. By not yet proven I mean: so far it has not been put to the (real) test. We have 18 reported bugs, and one can imagine to keep track of 18 in one's brain alone; only the future will tell how many bugs are reported . . .
- * **email:** In my eyes, email so far is not a problem, at least not the *official emails* via `swprakt+coma`, basically there's one or 2 emails per week. How groups use email internally, we don't exactly know.
- * **Face-to-face:** probably, in this phase, we need to use this more . . . Especially, we would like to recommend also to make separate appointments (with each group) apart from the general meetings. This will cost (also us) time, but is probably effective in the current stage.
- * **Web-page:** we maintain the web-page as server for information (keeping the handouts, specs., the announcement emails etc) I think, that's important.

The remarks about the bad means are, in some sense, more *psychological* than *technical*. one usually tend to feel "neglected" if someone changes something without announcement, or adds a "better" version of the file, and this is aggravated by the tendency *not to read* the Readme's etc. ("I understand what I want, but what the heck has XXX checked in here, well, I don't care/I don't understand, mine was good, too, I go ahead").

- **Structure of the repos:** Also here, it starts getting out of hand. we see signs of bad structure, for instance *names* as directory identifiers. It is good to have clean areas of responsibilities, but I tend to think it as not a good sign of structuring to see it reflected by name in the "code".³ It could again be some polite way of avoiding communication and current conflicts ("Let me do my stuff in "Alice" and so I leave you in peace in your directory in "Bob", I don't even look). If it works like this, it's ideal, but probably, our situation is not as clean.

²and besides the fact that we have to come to a conclusion, which is important . . .

³At the server side, it might be tolerable, but the "user version" should not have to deal with packages called "Alice" and "Bob".

Related to the above problems that we see with some of the means of communication, we must insist of **keeping the repos clean!** By this I mean

- no **garbage**/no **duplicates**/no **stale code**: it must for the people affected, be clear at (almost) each point, what is there, what is the current version that is relevant etc. Of course, it's impossible (and we should not even try) that everyone knows everything what's there all the time. But: if something affects others, those concerned must be "warned" in advance before something is added/changed etc. This warning need not be an email or long thread, it can mean *sitting together*
- clear separation of own source and third-party stuff (if we have some such).