
Understanding and Harnessing Conflict

John C. Thomas

IBM T. J. Watson Research Center
PO Box 704
Yorktown Heights, NY 10598 USA
truthtable@aol.com

Abstract

We argue that the over-avoidance of constructive conflict within groups actually contributes to destructive conflict between groups. We describe how this process contributes to normally "bad" HCI design and then how similar processes can contribute to serious violent conflicts. We describe Patterns generally and then offer some that can be used to help identify and use intra-group conflict as a resource and speculate how this may ultimately decrease inter-group conflict.

Author Keywords

Conflict, war, peace, Pattern Languages, in-group, out-group, socialization.

ACM Classification Keywords

H.5.m. Information interfaces and presentation.

General Terms

Economics, Human Factors, Legal Issues

Introduction

Human beings are quintessentially social animals and socialization begins at birth (if not before) and proceeds powerfully throughout life. This has always been clear; in ancient Greece, banishment was considered an equivalent criminal sentence with death. Even for hardened criminals, the most severe punishment is solitary confinement. More recently, the profound neurological and chemical basis of some of the effects of social interaction have been explicated [1].

Elsewhere [2], we have argued that the "normal" socialization process that occurs during development of products and services is a major, though unintentional cause of unnecessary complexity for users. Because the development team works together on common goals, they necessarily begin to share specialized common vocabulary, assumptions, and beliefs. Unfortunately, if "naïve" end users are not continually brought into this process, the resulting product and

service will inevitably have terms, concepts and mental models which are “obvious” to everyone on the team but far from obvious to everyone else. During any complex process such as developing a product or service, certain decisions are made and then seldom revisited. In fact, those who question the underlying and agreed upon assumptions are typically met with impatience or scorn.

Closed In-Groups

Development groups typically only last a few years. Company cultures exhibit a similar but stronger effect that can last for decades. It is not surprising then, that cultural effects associated with national boundaries, language groups, and religious affiliations can exert still stronger effects because they operate over much longer periods of time.

In some, if not most, cases, long-standing groups tend, not only to codify the knowledge that the group shares but develop processes and procedures to insulate themselves from alternative perspectives and from any knowledge that might force them to question their beliefs. An interesting example of this appeared many years ago in an episode of an American television show called *Candid Camera*. In this episode, when smokers put money in a vending machine, rather than receiving their cigarettes, the machine played a message telling them how bad smoking was for their health. When the crew member of *Candid Camera* approached and asked them what happened, they universally reported that the machine had not given them their cigarettes but instead had played an *advertisement for cigarettes*. They failed to process the actual contents of the message. In similar fashion, those who hold prejudiced views of another religious or ethnic group do not

generally “revoke” their beliefs when they encounter contradictory evidence but instead explain it away. Is it possible to imagine groups which reap the social benefits of being “close-knit” while at the same time encouraging encounters with differing ideas, perspectives, and experiences?

Pattern Languages.

We believe an approach to answering this question affirmatively may be found in the realm of a socio-technical Pattern Language. A Pattern is the named outline solution to a recurring problem along with an analysis of that problem. A Pattern Language is a lattice of interconnected Patterns that attempts to deal with an entire domain. This notion was introduced by Christopher Alexander [3] and his colleagues in the realm of urban planning and architecture. Since then it has been applied in many other domains including object-oriented software architecture [4], human-computer interaction [5], and social change [6].

We propose that there are socio-technical Patterns that can simultaneously make groups operate more effectively and efficiently *and* at the same time, make them more open to understanding and constructively using input from others who have diverse experiences, cultures, beliefs, and languages. An example of such a Pattern is “Who Speaks for Wolf?” [7] which is intended to insure that a group includes all major viewpoints and concerns relevant to a problem early on. It is our contention, for example, that this is good not only for the “in-group” but also makes the “in-group” more amenable to diverse viewpoints from other groups.

Sample Pattern: Greater Gathering

Another example of such a Pattern is Greater Gathering. This basically says that for work to be accomplished effectively, the overall work is most often broken down hierarchically into sub-goals and associated sub-groups. However, these sub-groups begin to develop their own vocabularies and interests. Greater Gathering suggests that it is very productive to periodically get everyone back together. Exactly how this is done varies according to circumstance but some existing examples include Company Picnics, Scout Jamborees, Celebrations and Parades, CHI, and the Olympics.

Support for Conversation Across Boundaries

This Pattern was originally conceived as useful for sub-groups within an organization but the same principle applies at the level of communicating across national, religious, and language boundaries.

References

- [1] Goleman, D. *Social Intelligence: The New Science of Human Relationships*. New York: Bantam, (2006).
- [2] Thomas, J. C. & Richards, J. R. . Achieving psychological simplicity: Measures and methods to reduce cognitive complexity. In *Handbook of HCI*, A. Sears & J. Jacko (Eds.). Hillsdale, NJ: Erlbaum (2008).
- [3] Alexander, C., Ishikawa, S., and Silverstein, M. (1977), *A Pattern Language*. New York: Oxford University Press.
- [4] Gamma, E., et al. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading, MA, 1995.

Sample Pattern: Small Successes Early.

This Pattern does not deal with the structure of organizations but with a process. In attempting any complex undertaking with people who do not already have a working relationship, it is useful to begin, not by directly tackling the entire problem but by tackling a fairly simple sub-problem or even by working together on something else manageable. In this way, people can get to understand each other in a less complex and stressful situation before attempting the more complex issues. Actual Patterns typically include pictures, an analysis of the problem, examples, and references. Space does not permit this here, but more complete examples can be found in [6] and [7]. I would like to present the Pattern Language approach along with some examples and get feedback from others in the workshop about its potential utility as well as learn about completely different approaches that others are taking.

- [5] Bayle, E., Bellamy, R., Casaday, G., Erickson, T., Fincher, S., Grinter, B., Gross, B., Lehder, D., Marmolin, H., Potts, C., Skousen, G. & Thomas, J. Putting It All Together: Towards a Pattern Language for Interaction Design. Summary Report of the CHI '97 Workshop. SIGCHI Bulletin. New York: ACM,
- [6] Schuler, D. (2008), *Liberating voices: A pattern language for communication revolution*. Cambridge, MA: MIT Press.
- [7] Thomas, J.C., Lee, A., & Danis, C (2002). "Who Speaks for Wolf?" *IBM Research Report*, RC-22644. Yorktown Heights, NY: IBM Corporation.

