Cooperative Development Inside Communities

Jeff Osier-Mixon

MontaVista Software, Inc.

Embedded Linux Conference 2009
Kabuki Hotel, San Francisco, CA

Who is Jeff?

- Veteran technical writer
- Embedded open-source greybeard: has worked in embedded open-source software and related hardware for 17 yrs (OMG)
- Experienced web guru: created one of the first corporate websites and intranets in 1994 (and hasn't stopped since)
- Longtime community volunteer in many areas: homeschooling, airport, Habitat, historical society
- Developer advocate and open-source blogger at http://www.jefro.net/blog

Why is he talking to us?

- Passionate about community and its potential for research efficiency, developer harmony, and world peace
- Sees something missing in the Linux community and wants to help
- Admin for MontaVista's new developer community

Who is MontaVista Software?

- Leader in embedded software solutions based on Linux
- Major developer of real-time features
- Major innovator and code submitter to mainline
- A company of developer advocates

Community-Oriented Software Development

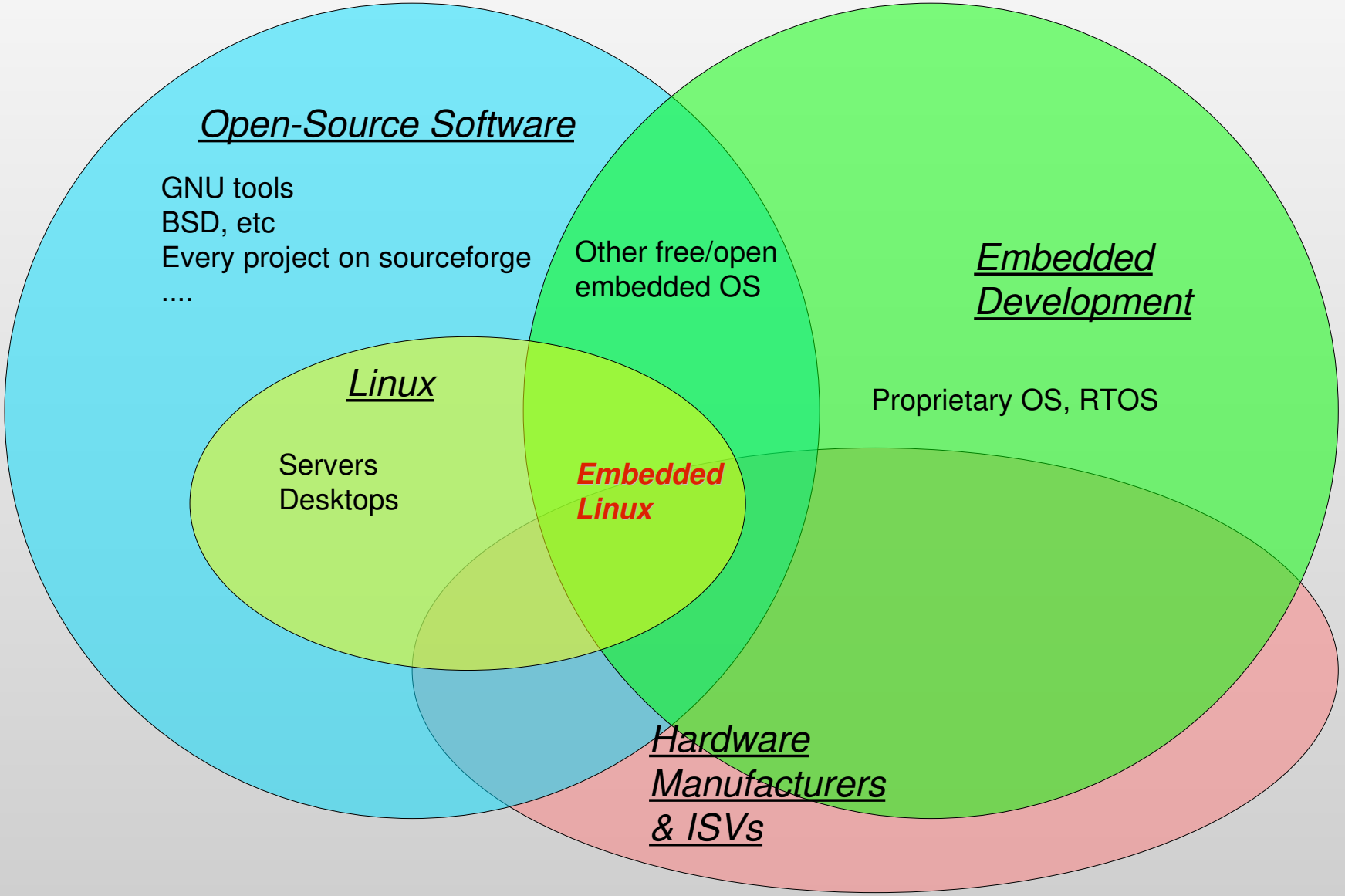
The Benefits of Community-Based Development

- Preaching to the converted

The Caveats of Community-Based Development

- Still preaching to the converted

Who is the Community?

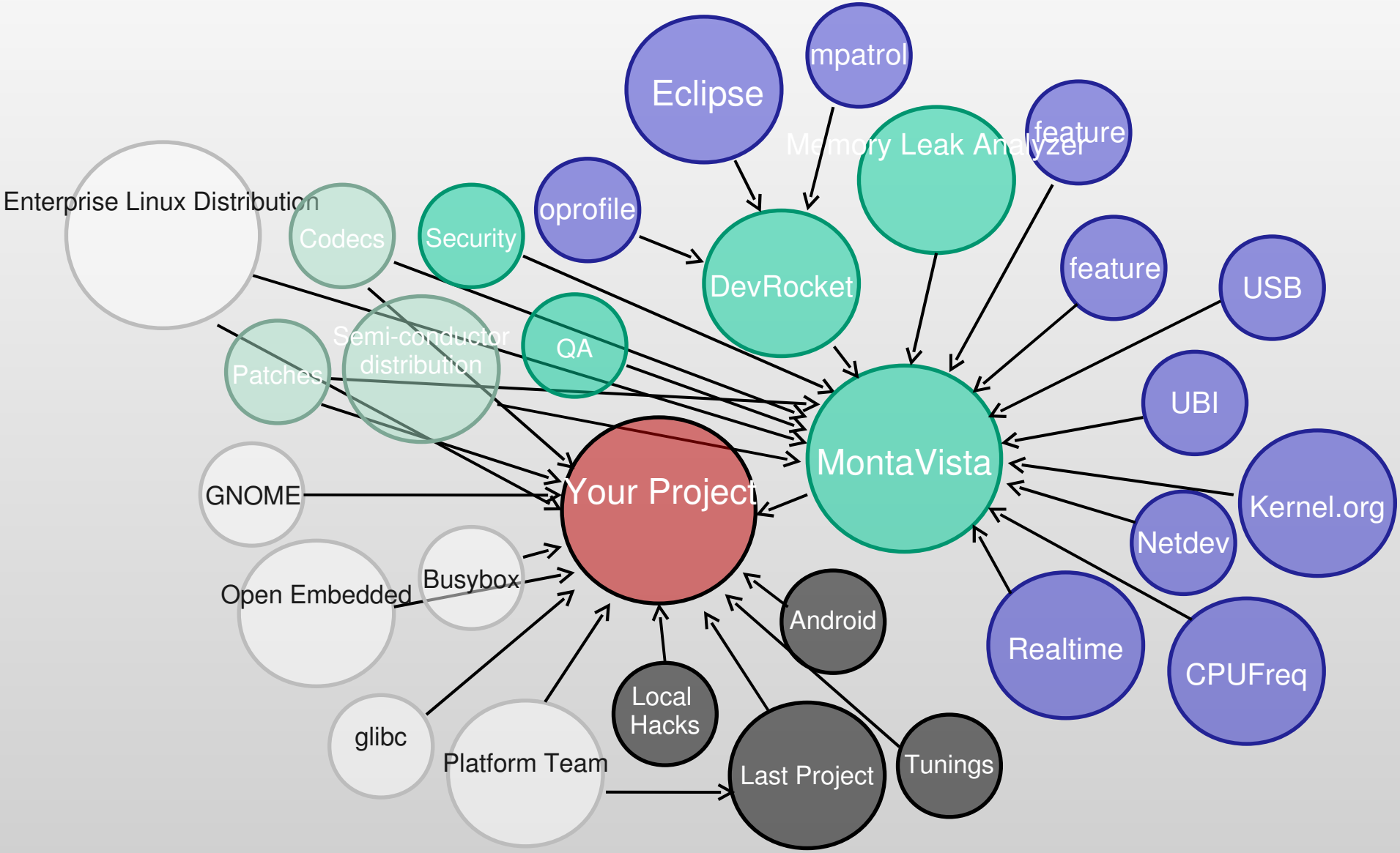


Let's Look Closer at the Embedded Linux Community and its challenges

Because it's not “the” community you are after.

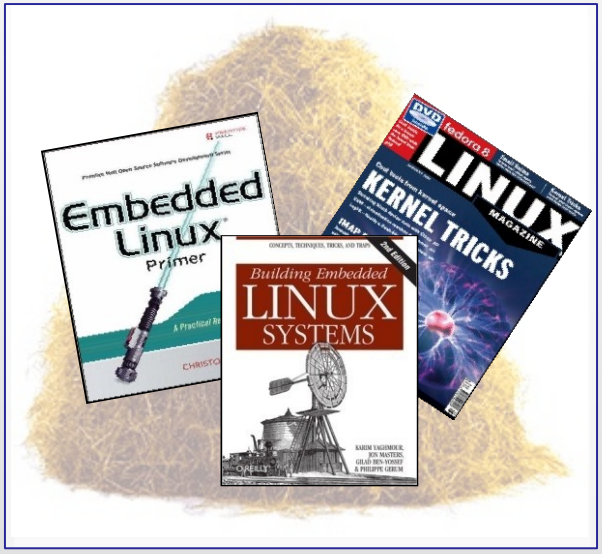
It is YOUR community you want---the one that answers your questions and enables your participation

One View of the Embedded Linux Community



© 2009 MontaVista Software

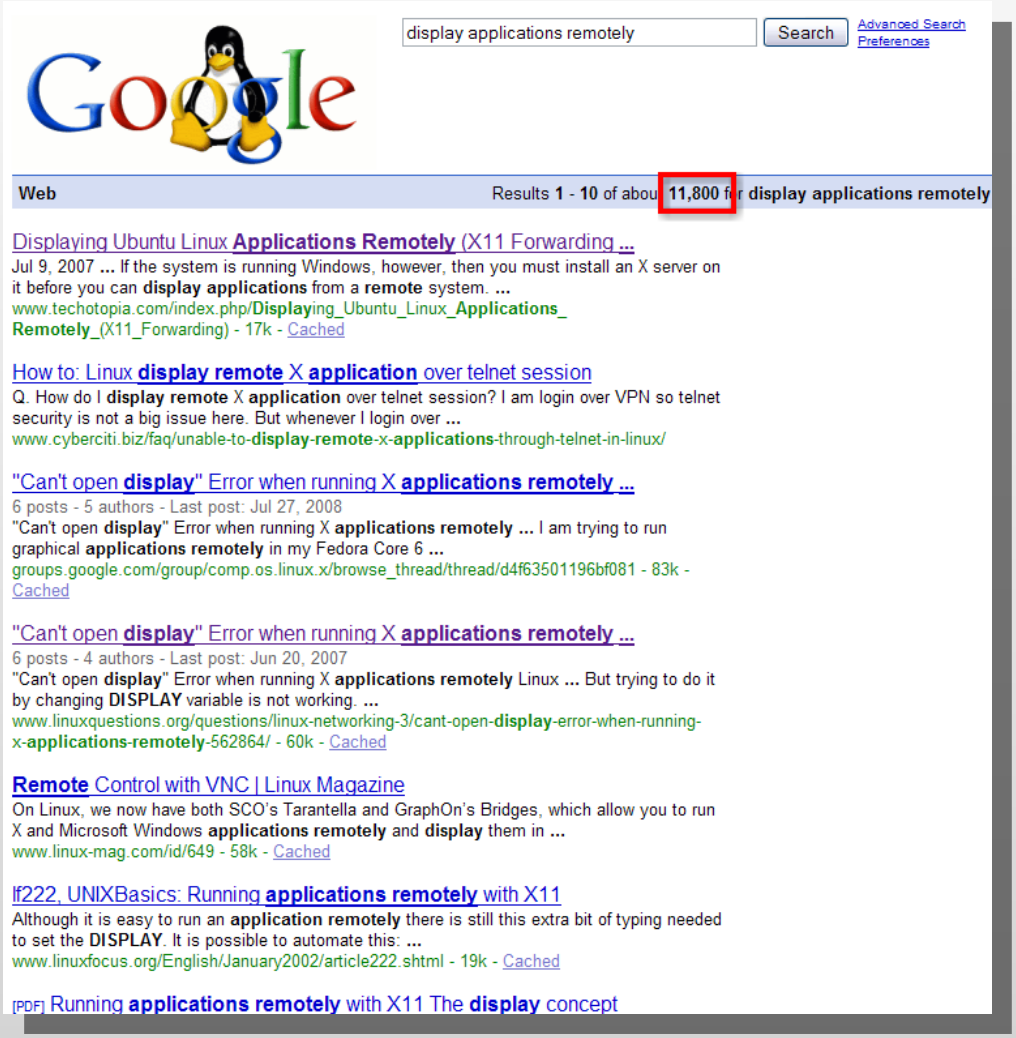
Finding a Needle in Multiple Haystacks



© 2009 MontaVista Software

People, Not Pages

Vs.



Over 12,000 results

© 2009 MontaVista Software

The Challenges

Embedded Linux development is information intensive

Locate information, hardware and software

Validate information, credibility and completeness

Adapt information, specific project needs

© 2009 MontaVista Software

A Proposed Solution

An embedded Linux community that connects all engineers

Connect: **Leverage the experience of other embedded Linux developers**

Share: **Collaborate on information and ideas**

Design: **Develop products faster with information re-use**

© 2009 MontaVista Software

Embedded Linux community,
For systems engineers,
To connect and share information, ideas, and software,
Unlike search engines, Meld directly connects experienced systems engineers to assist one another on specific projects and to design commercial-ready embedded devices.

© 2009 MontaVista Software

Meld…

IS	IS NOT
Community of embedded Linux users	An open source project for code development
Supporting commercial device development	Targeting academic, research, hobbyist development
Connecting embedded developers who use Linux, connecting communities	Replacement for already existing communities
Embedded: device-specific root filesystem	Enterprise/desktop: one-size-fits-all distribution
Public, open	MontaVista customers only, paid for, application specific

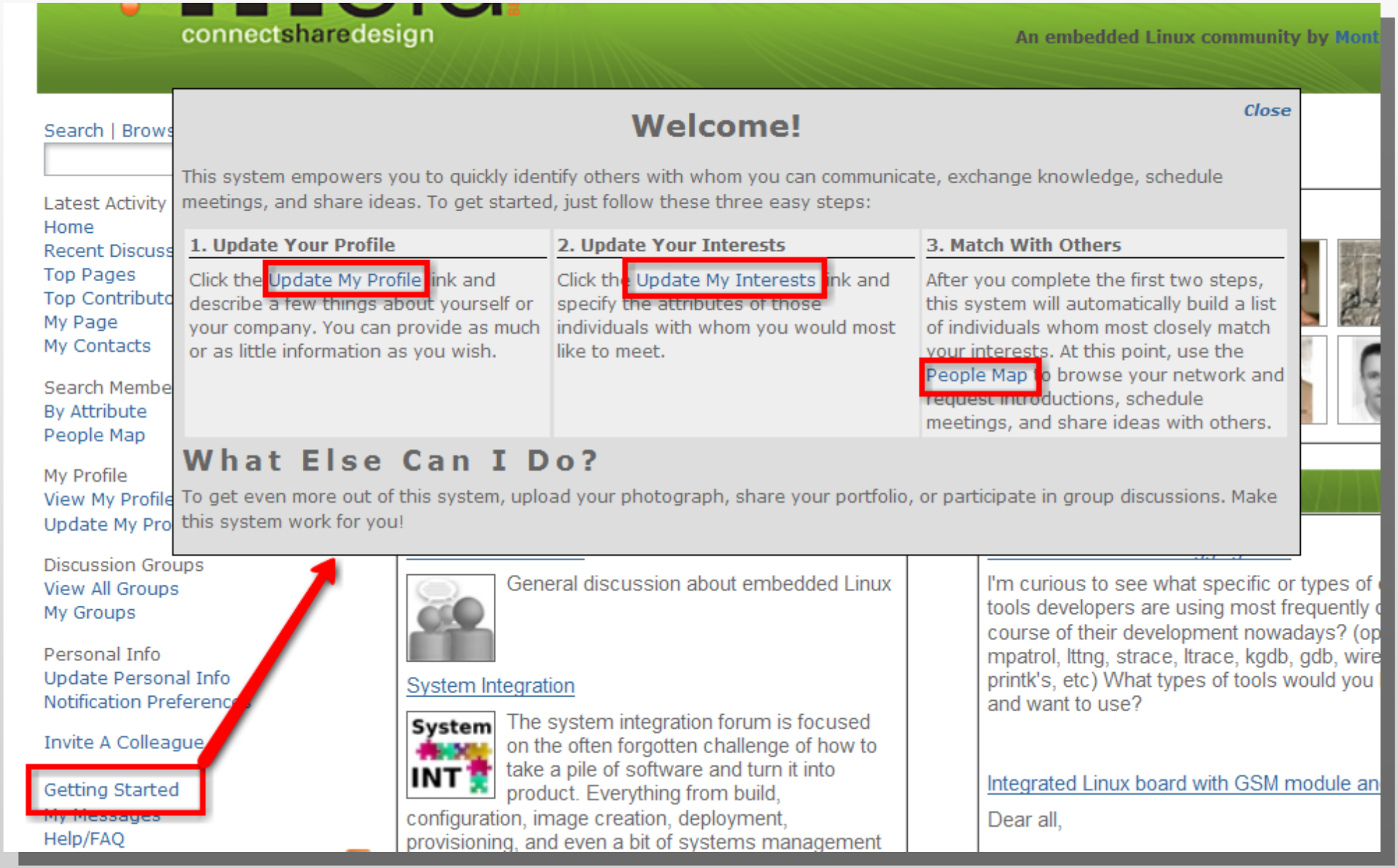
© 2009 MontaVista Software

Welcome to Meld

meld.mvista.com

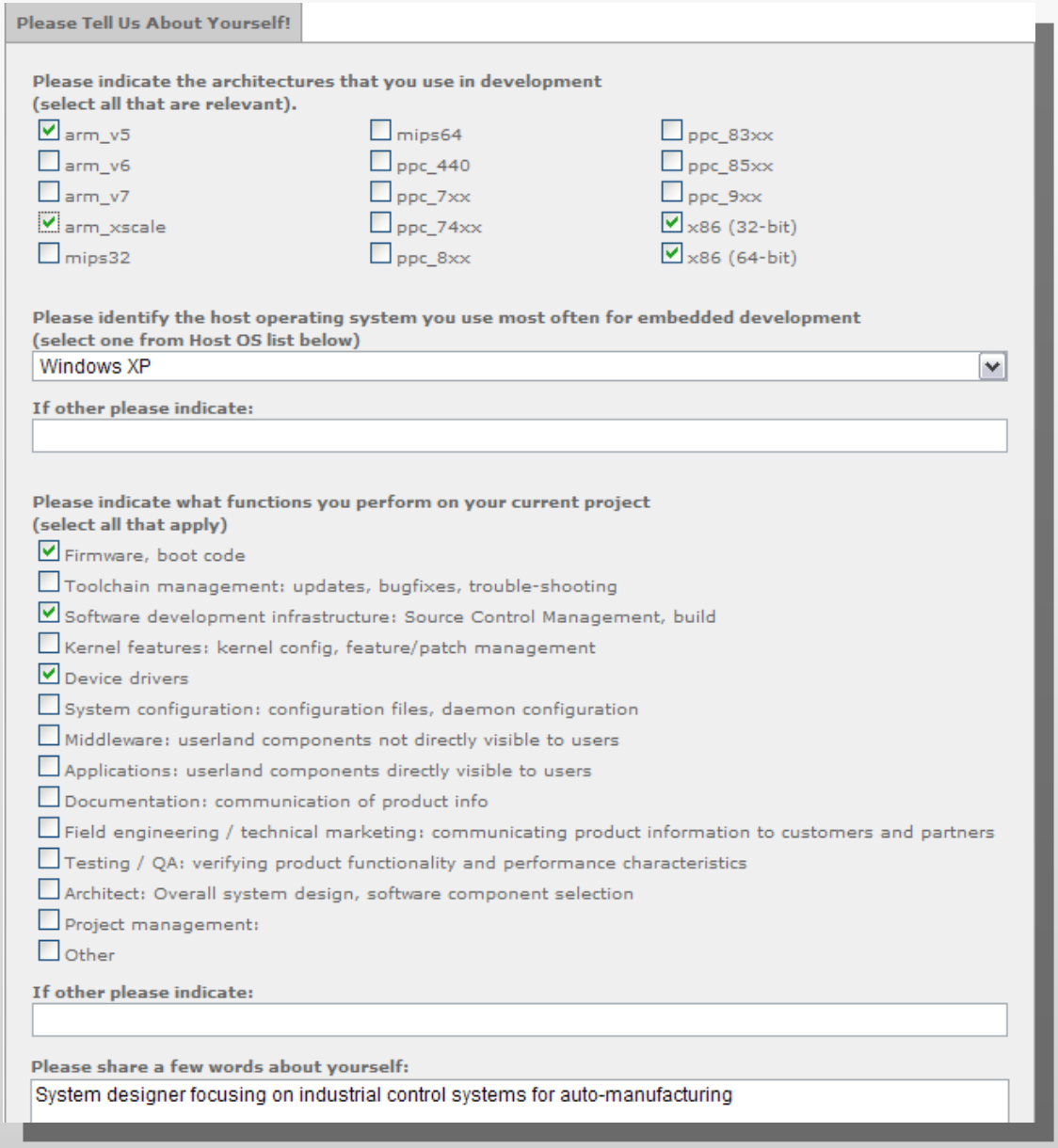
© 2009 MontaVista Software

Connect: Update My Interests



© 2009 MontaVista Software

Connect: Update My Profile



© 2009 MontaVista Software

Connect: Update My Interests

Who Would You Like to Meet?

I would like to meet members performing the following functions:

- ☐ Firmware, boot code
- ☐ Toolchain management: updates, bugfixes, trouble-shooting
- ☐ Software development infrastruction: Source Control Mangement, build
- ☐ Kernel features: kernel config, feature/patch management
- ☐ Device drivers:
- ☐ System configuration: configuration files, daemon configuration
- ☐ Middleware: userland components not directly visible to users
- ☐ Applications: userland components directly visible to users
- ☐ Documentation: communication of product info
- ☐ Field engineering / technical marketing: communicating product information to customers and partners
- ☐ Testing / QA: verifying product functionality and performance characteristics
- ☐ Architect: Overall system design, software component selection
- ☐ Project management:
- ☐ Other

I would like to meet members using these architectures for development. (If your architecture is not shown, please let us know at community@mvista.com.)

☐ arm_v5	☐ ppc_440	☐ ppc_85xx
☐ arm_v6	☐ ppc_7xx	☐ ppc_9xx
☐ arm_xscale	☐ ppc_74xx	☐ x86 (32-bit)
☐ mips2	☐ ppc_8xx	☐ x86 (64-bit)
☐ mips64	☐ ppc_83xx	

With the following Host OS

☐ Fedora Linux	☐ Other Linux	☐ Solaris 8
☐ RedHat Linux	☐ Windows XP	☐ Solaris 9
☐ Suse Linux	☐ Windows 2000	☐ Other

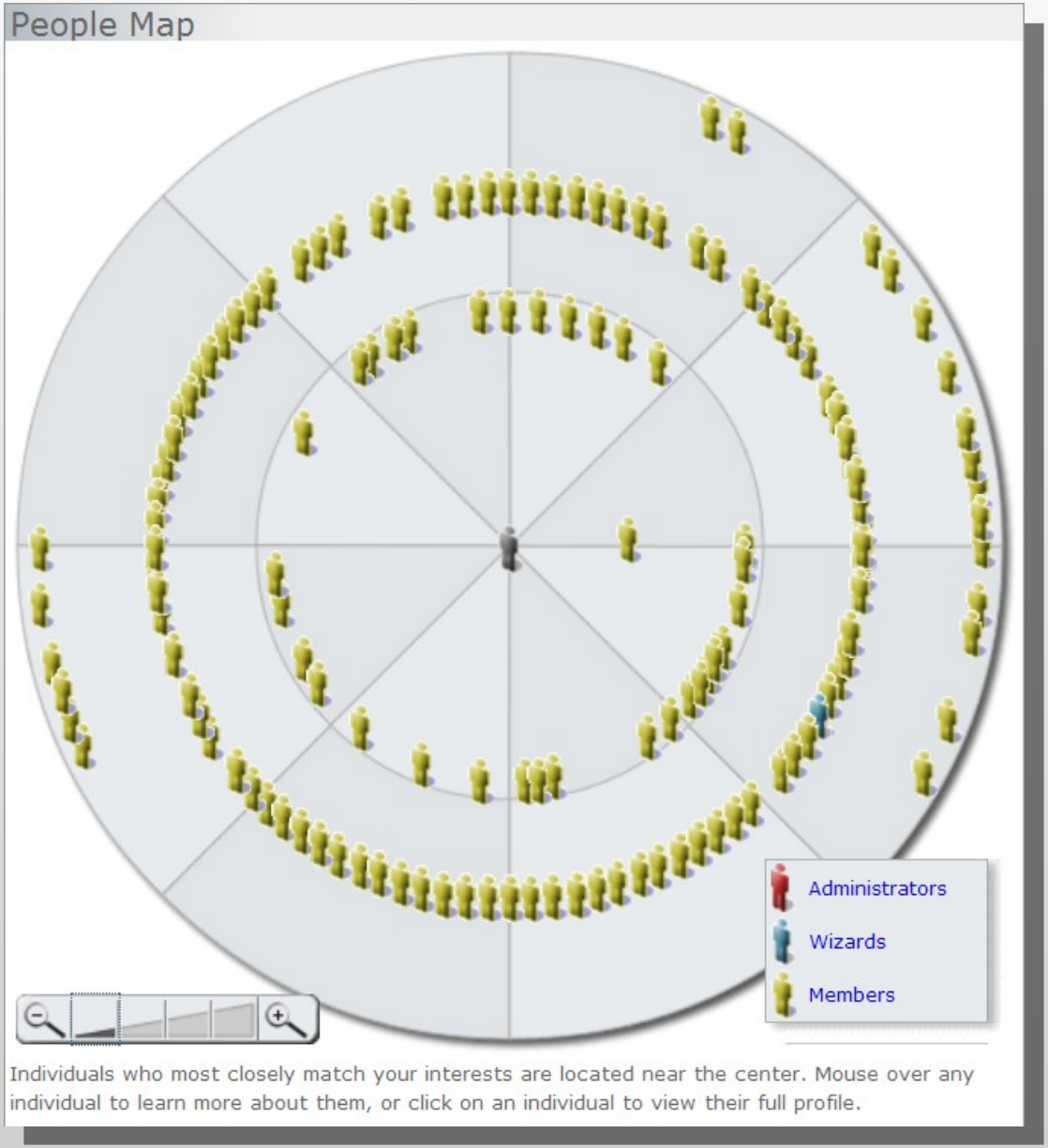
What are your top objectives for joining the community?

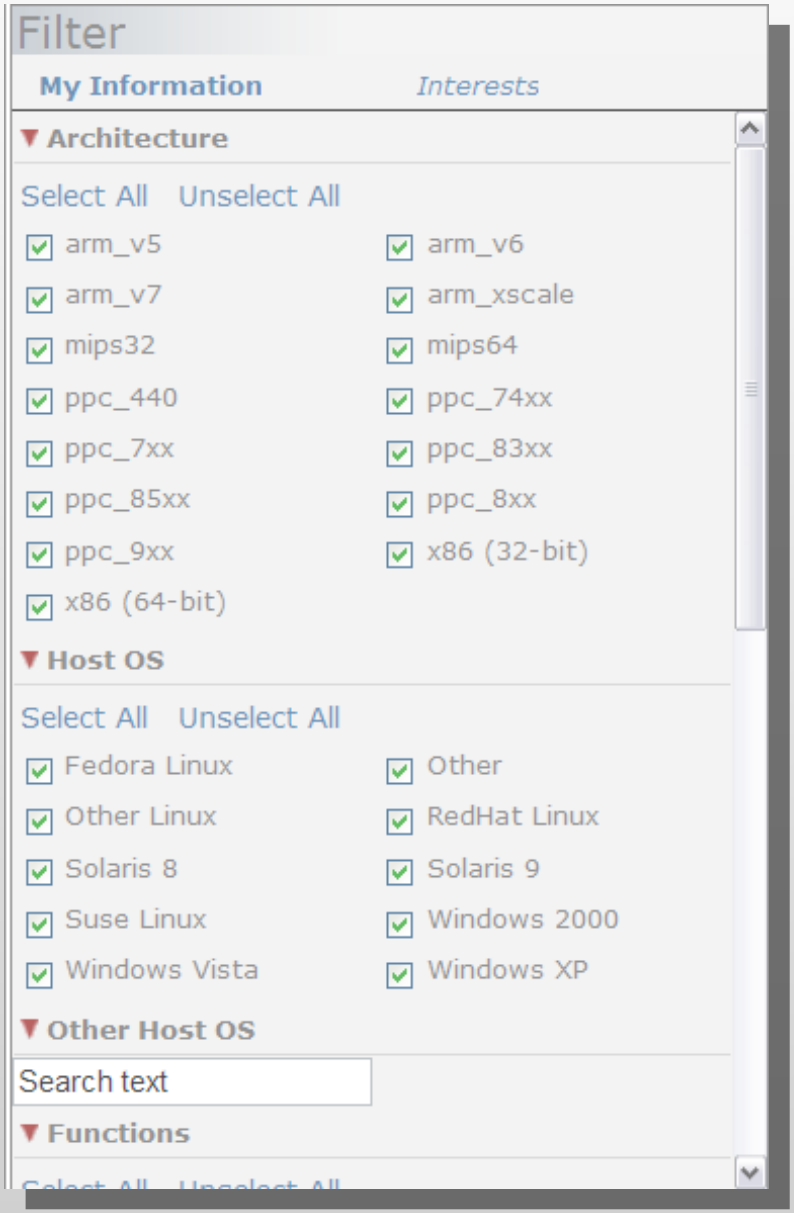
© 2009 MontaVista Software

Connect: Update My Interests

© 2009 MontaVista Software

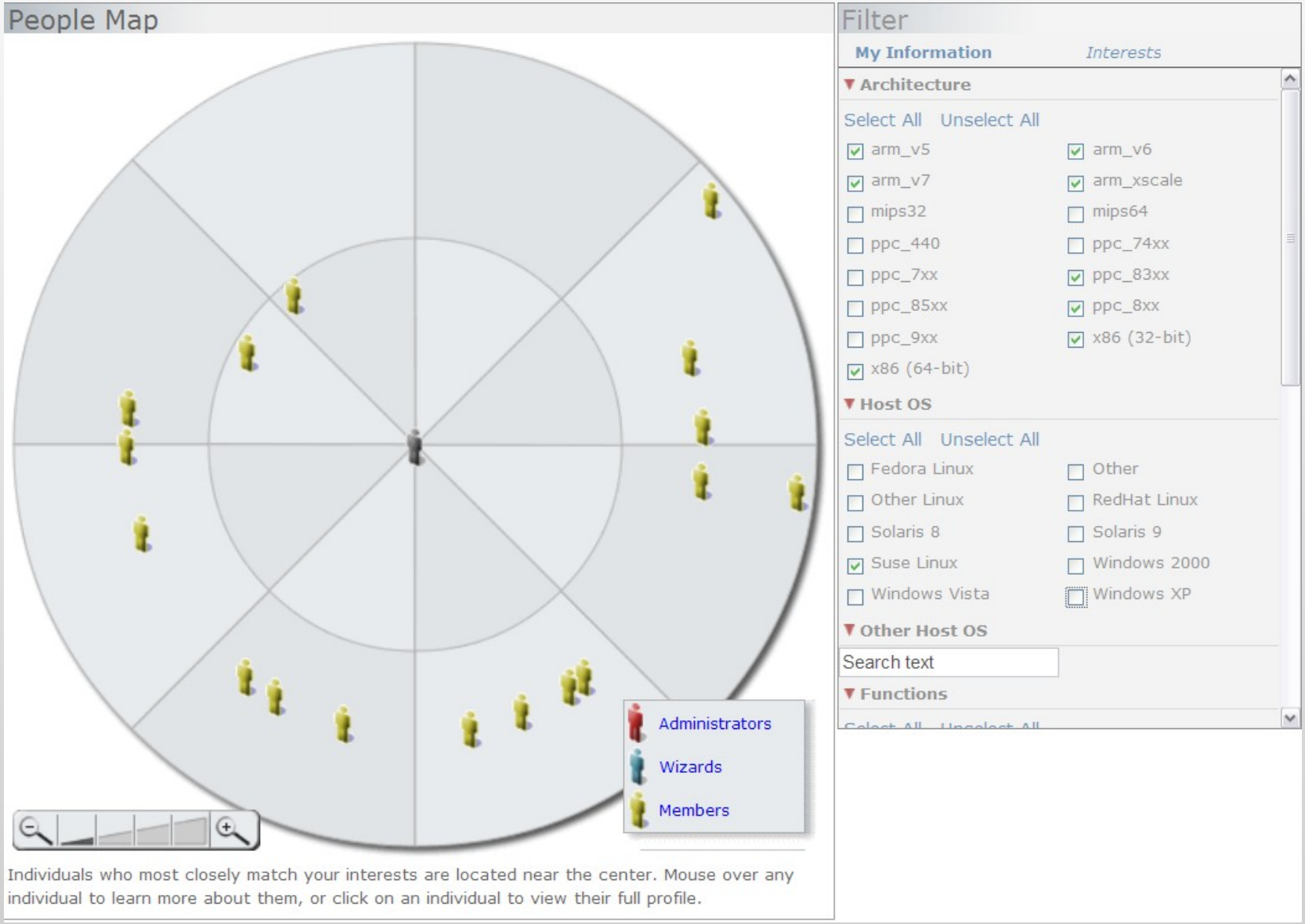
Connect: People Map





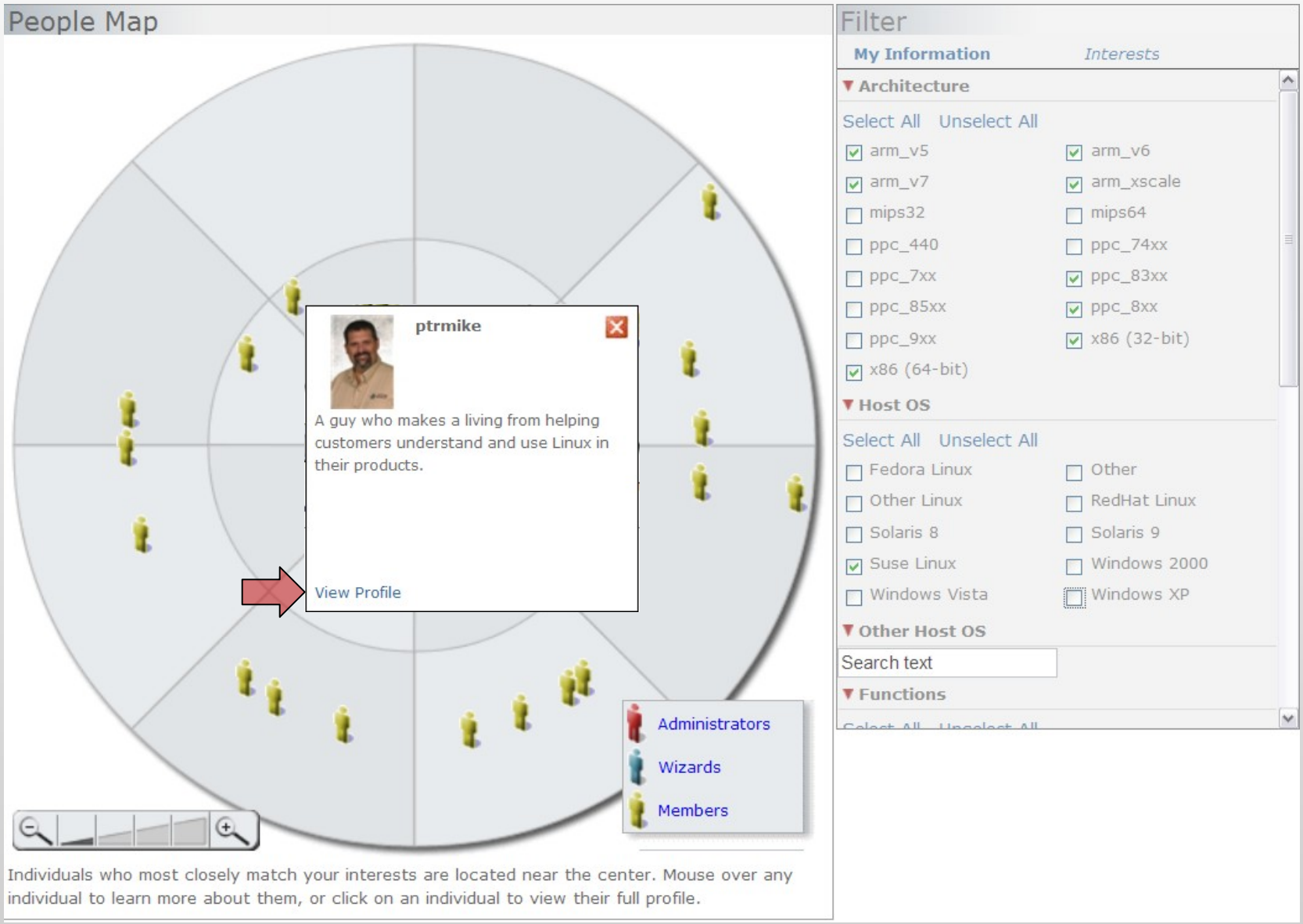
© 2009 MontaVista Software

Connect: People Map Filter



© 2009 MontaVista Software

Connect: View Profile



© 2009 MontaVista Software

Connect: User Profiles

ptrmike

Member since February 27, 2009 | My Activity

Top Contributor

Total Contributions 28

Send a message

Request a meeting

Add to contacts

ABOUT ME

Architecture

arm_v5, arm_v6, arm_v7, arm_xscale, mips32, mips64, ppc_440, ppc_8xx, ppc_83xx, ppc_85xx, x86 (32-bit), x86 (64-bit)

Host OS

Other Linux

Other Host OS

Ubuntu

Functions

Firmware, boot code, Toolchain management: updates, bugfixes, trouble-shooting, Software development infrastructure: Source Control Management, build, Kernel features: kernel

About Me

A guy who makes a living from helping customers understand and use Linux in their products.

I Am Looking For

To be able to see what others are doing to solve the same problems that I am. We're all in the same boat and somebody surely has an idea of which end of the oar to grab.

My Community Activity

async. notofication using signals from kernel space to userland

Yes, spawn a thread that blocks on a read in your driver. Then, have the interrupt, timer, whatever do a wake_up_interruptible to the driver to release the blocked read. That should be much faster than using more...

17 hours ago |

System integration webinar

Although I certainly understand David Woodhouse's perspective on always having the latest kernel, I feel it neglects the reality of manufacturing and QA. Any change that you make to a product's software more...

3/19/2009 3:43 PM | 3

Polling GPIO pin status

Hmm... You'll need to be more specific. What CPU architecture are you using? Are the GPIO lines I/O mapped or memory mapped? You say you have a user-space "driver". How did you implement it? Using more...

3/19/2009 8:26 AM | 1

Linux on emerson MVME3100 board

It could be done any of the ways you've outlined. Since I'm not from Emerson and don't have access to the code, I really couldn't say how they were doing it. I'm sure you'll be able to access the bus from the more...

3/18/2009 7:42 AM | 1

Firmware loading using firmware.agent

The firmware is typically located in /lib/firmware. The firmware will be loaded when your driver does the request_firmware call. And, did you look in at the example in <linux>/samples/firmware_class? more...

© 2009 MontaVista Software

Connect: Email, Meetings

© 2009 MontaVista Software

Connect: Learn More About Melders

© 2009 MontaVista Software

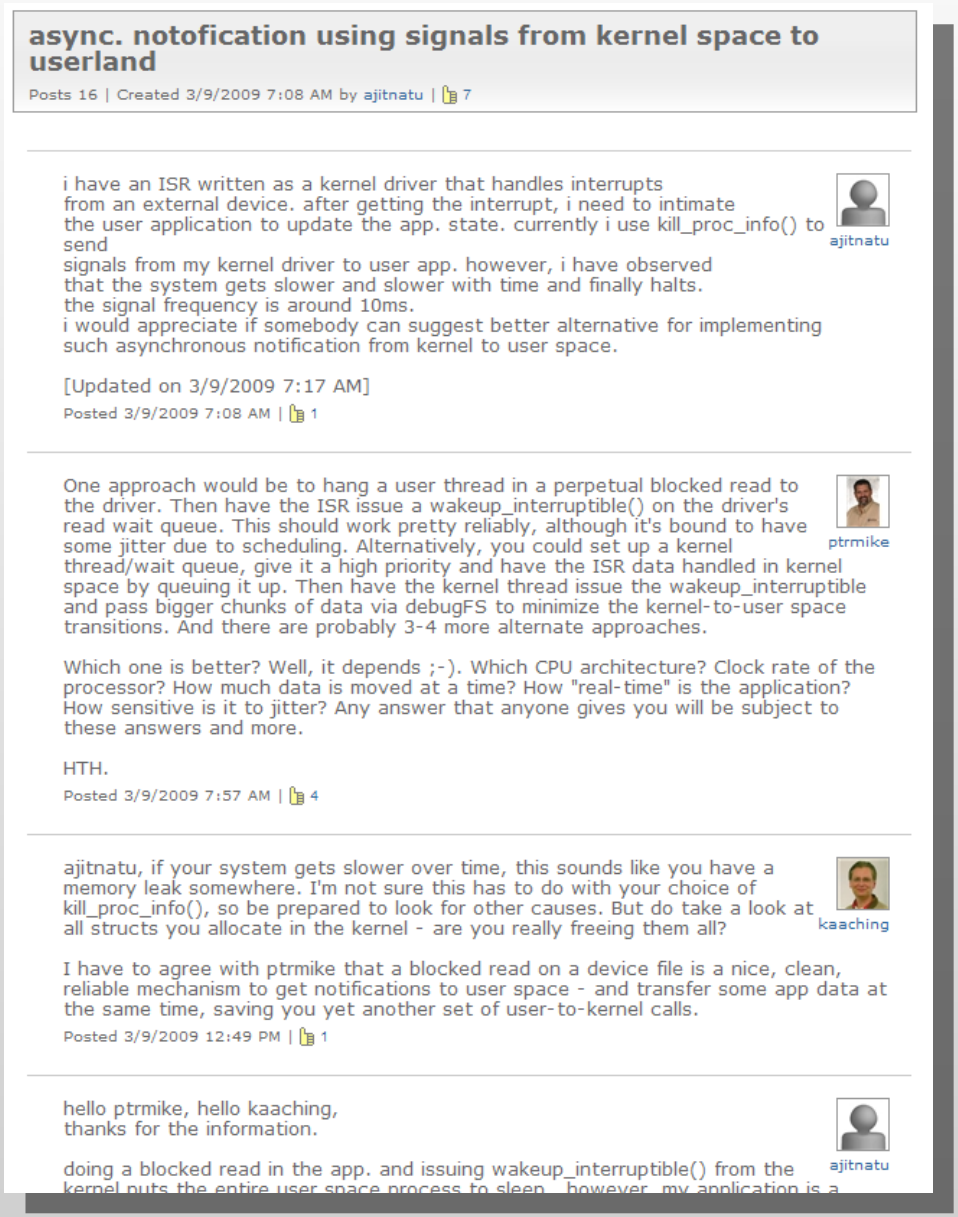
Connect: Learn More About Melders

© 2009 MontaVista Software

Share: Group Discussions

© 2009 MontaVista Software

Share: Group Discussions



async. notofication using signals from kernel space to userland

Posts 16 | Created 3/9/2009 7:08 AM by ajitnatu | 7

ajitnatu

i have an ISR written as a kernel driver that handles interrupts
from an external device. after getting the interrupt, i need to intimate
the user application to update the app. state. currently i use kill_proc_info() to send
signals from my kernel driver to user app. however, i have observed
that the system gets slower and slower with time and finally halts.
the signal frequency is around 10ms.
i would appreciate if somebody can suggest better alternative for implementing
such asynchronous notification from kernel to user space.

[Updated on 3/9/2009 7:17 AM]

Posted 3/9/2009 7:08 AM | 1

ptrmike

One approach would be to hang a user thread in a perpetual blocked read to the driver. Then have the ISR issue a wakeup_interruptible() on the driver's read wait queue. This should work pretty reliably, although it's bound to have some jitter due to scheduling. Alternatively, you could set up a kernel thread/wait queue, give it a high priority and have the ISR data handled in kernel space by queuing it up. Then have the kernel thread issue the wakeup_interruptible and pass bigger chunks of data via debugFS to minimize the kernel-to-user space transitions. And there are probably 3-4 more alternate approaches.

Which one is better? Well, it depends ;-). Which CPU architecture? Clock rate of the processor? How much data is moved at a time? How "real-time" is the application? How sensitive is it to jitter? Any answer that anyone gives you will be subject to these answers and more.

HTH.

Posted 3/9/2009 7:57 AM | 4

kaaching

ajitnatu, if your system gets slower over time, this sounds like you have a memory leak somewhere. I'm not sure this has to do with your choice of kill_proc_info(), so be prepared to look for other causes. But do take a look at all structs you allocate in the kernel - are you really freeing them all?

I have to agree with ptrmike that a blocked read on a device file is a nice, clean, reliable mechanism to get notifications to user space - and transfer some app data at the same time, saving you yet another set of user-to-kernel calls.

Posted 3/9/2009 12:49 PM | 1

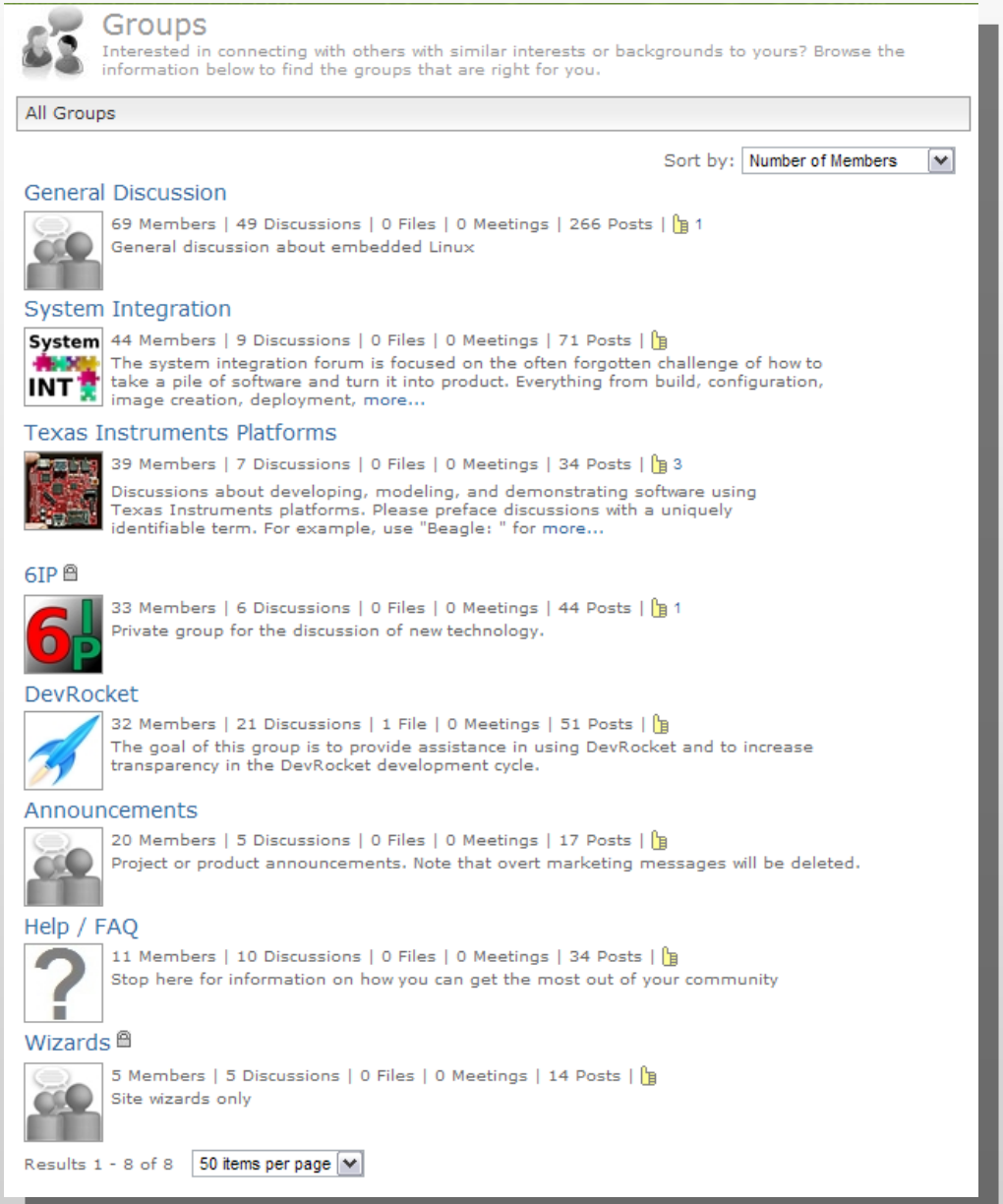
ajitnatu

hello ptrmike, hello kaaching,
thanks for the information.

doing a blocked read in the app. and issuing wakeup_interruptible() from the kernel puts the entire user space process to sleep. however, my application is a

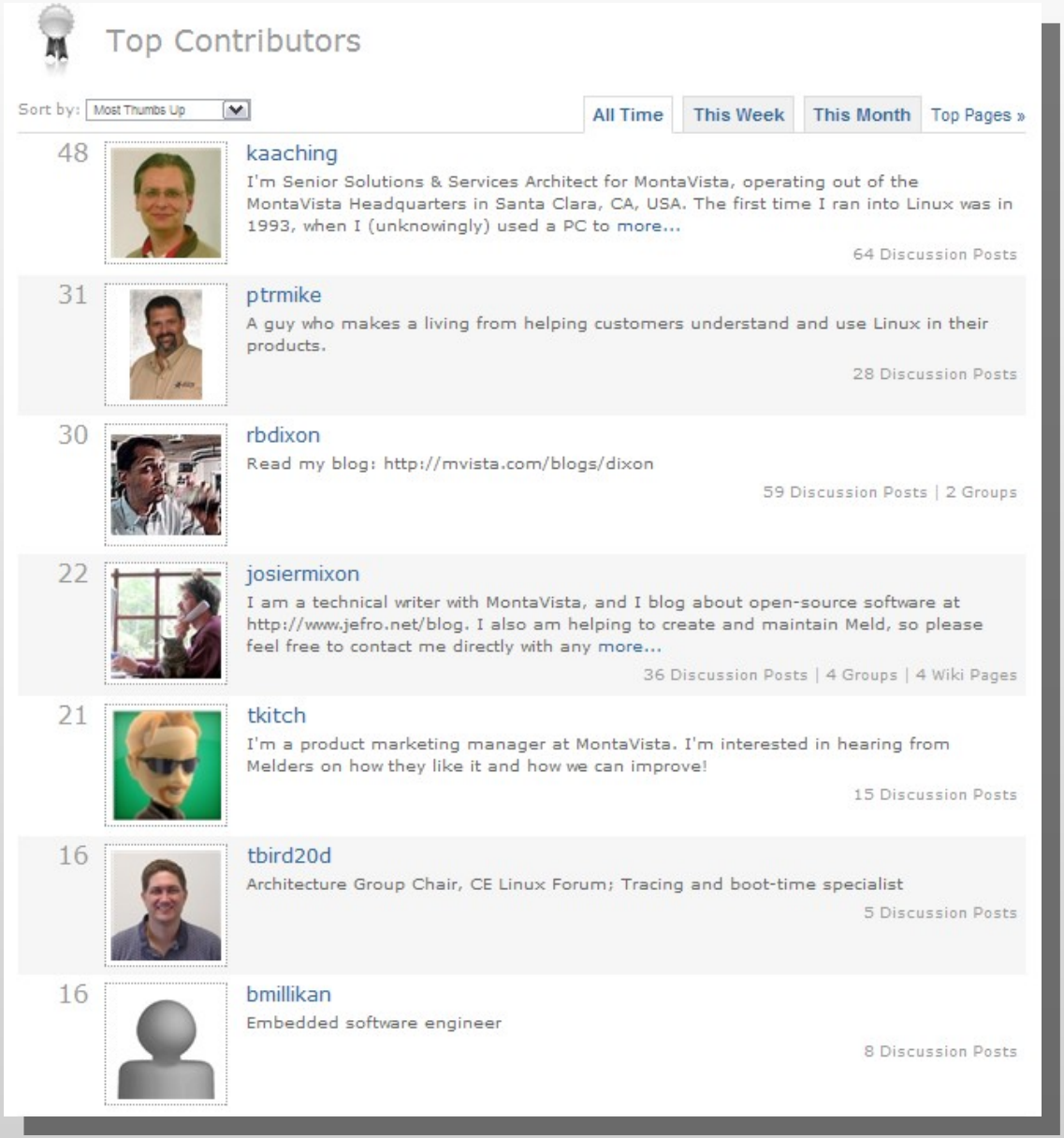
© 2009 MontaVista Software

Share: Groups



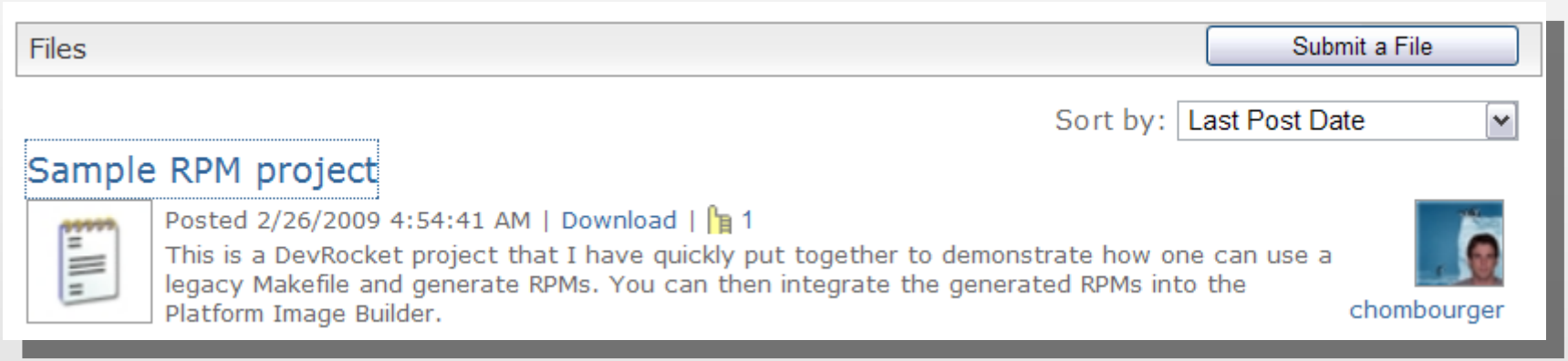
© 2009 MontaVista Software

Share: Top Contributions

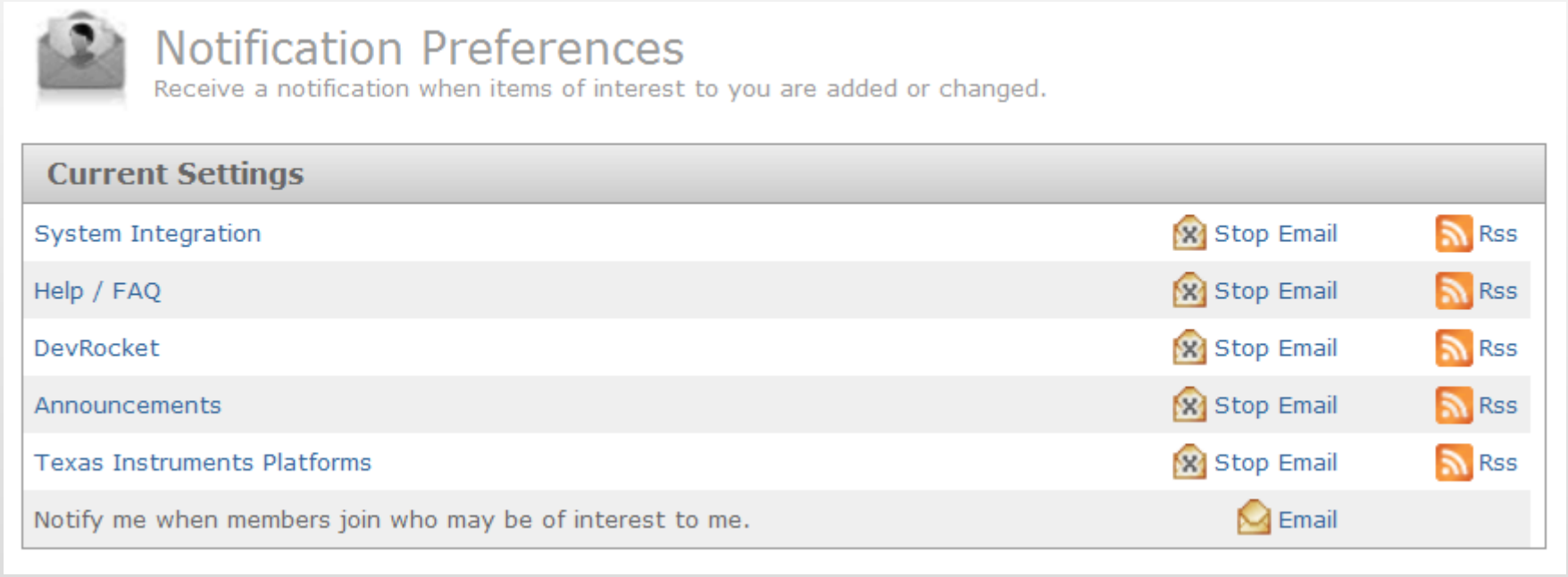


© 2009 MontaVista Software

Share: Content, Files



Share: Notification Preferences



Design

Developer Spotlight

Kurt Lloyd
Alcatel-Lucent

"Meld is a well-implemented framework for facilitating effective communication among people using (or wanting to use) open source in their products. Good job putting it together!"

Kurt has worked on a variety of high-end telecommunication products, with more to come. Kurt is currently working on a large DWDM switch, and some smaller switches using much of the same code base.

Why is MontaVista doing this?

- As a company, MontaVista realizes that the best return on investment comes from happy developers
- The end goal of Meld is to create happy developers
- The beyond-the-end goal is happy ISVs and hardware manufacturers, who will create more and better products for those happy developers
- MontaVista wants to foster the communication to make this possible---collaboration is king

Thanks for participating!

meld.mvista.com

994 members, be the next one