محمود فتحی

ارائه الگوریتمی برای دسته‌بندی بسته‌ها در شبکه اینترنت.

محمد قدسی

برنامه‌ریزی حرکت و برچسب گذاری اشیاء متحرک.

شهره کسائی

روش موثر برای تائید هویت افراد با استفاده از تصاویر اثر انگشت.

مرتضی منیری

ـ معناشناسی جهان‌های ممکن برای منطق موجهات مرتبهٔ اول،

ـ ارتباط بین تئوری پیچیدگی و منطق.

علی موقر رحیم آبادی

طراحی و تحلیل شبکه‌های کامپیوتری سریع.

محمدرضا میبدی

ـ آتوماتای یادگیر سلولی: یک مدل ریاضی برای مسائل پیچیده و غیرقطعی،

ـ حل مسأله کوتاه‌ترین مسیر توسط آتوماتای یادگیر،

ـ آتوماتای سلولی یادگیر بسط داده شده.

قاسم میرعمادی

ارزیابی سیستم‌های اتکاء پذیر مبتنی بر مدارات برنامه‌ریز با استفاده از تزریق خطا در فایل پیکربندی.

محمد حسین یغمائی مقدم

طراحی و پیاده‌سازی یک الگوریتم فازی برای مدیریت توام صف و طراحی و پیاده‌سازی مکانیزم زمانبندی فازی برای شبکه‌های محلی بدون سیم.

******

در این شمارهٔ **اخبار**، صفحاتی را به پژوهشکدهٔ علوم کامپیوتر پژوهشگاه و مباحثی در حول‌وحوش فعالیت‌های آن اختصاص داده‌ایم. در مطلب بالا با این پژوهشکده و برنامه‌ها و عملکردش آشنا شدید. در ادامه، مقالهٔ «آشنایی با روش‌های صوری در مهندسی نرم‌افزار» را می‌خوانید که موضوع آن از مباحث مهم روز در علوم کامپیوتر است و در ضمن از موضوعات اصلی همایشی بود که این پژوهشکده سال گذشته در زمینهٔ مهندسی نرم‌افزار برگزار کرد. سپس دو مقاله دربارهٔ ابررایانه و ابررایانش می‌آید. پژوهشکدهٔ علوم کامپیوتر مشغول مطالعهٔ امکان‌سنجیِ ساخت ابررایانش در ایران است. مراکز دیگری نیز در کشور ما، هر یک از دیدگاهی و با در نظر داشتن کاربردهایی، به این مقوله توجه دارند. از این‌رو درج این دو مقاله که خواننده را با مبانی این موضوع و فعالیت‌های جاری در این زمینه در سطح جهانی آشنا می‌سازند، می‌تواند سودمند افتد. و بالاخره، این مجموعه را با گزارشی تفصیلی از «یازدهمین کنفرانس کامپیوتر ایران» به پایان می‌آوریم.

آشنایی با روش‌های صوری در مهندسی نرم‌افزار

مصطفی زالی، مرجان سیرجانی**

اهمیت مهندسی نرم‌افزار

در سال ۱۹۵۰، هنگام پیدایش نخستین نرم‌افزارها، کسی گمان نمی‌کرد که روزی نرم‌افزار به یک تکنولوژی ضروری برای علوم پایه، مهندسی و تجارت تبدیل شود. امروزه نرم‌افزار امکان ایجاد تکنولوژی‌های جدید مانند مهندسی ژنتیک، گسترش صنایع موجود از قبیل مخابرات و منسوخ کردن تکنولوژی‌های قدیمی چون صنعت چاپ را فراهم کرده است. علاوه بر این نقش نرم‌افزار در تنظیم مناسبات فرهنگی و اقتصادی غیر قابل انکار است. برای درک جایگاه مهندسی نرم‌افزار و اهمیت آن در مناسبات اقتصادی و صنعتی می‌توان به آمار وضعیت نرم‌افزار در سال ۱۹۹۶ در ایالات متحدهٔ آمریکا مراجعه کرد: در سال ۱۹۹۶ نرم‌افزار بخش قابل توجهی از میزان سود خالص را در میان اقلام صادراتیِ این کشور به خود اختصاص داد که بالغ بر ۲۰ میلیارد دلار برآورد شده است؛ این میزان برای محصولات کشاورزی، صنایع فضایی، شیمیایی و وسایل نقلیه به ترتیب ۱۲، ۸، ۷ و ۲۲ میلیارد دلار بوده است.

مشکلات ناشی از اشتباه در نرم‌افزار

با توجه به این مطالب می‌توان دریافت که با رواج استفاده از نرم‌افزار در امور تجاری، صنعتی، آموزشی و ... میزان خسارات حاصل از خطاهای نرم‌افزاری ممکن است هزینه‌هایی غیرقابل چشم‌پوشی برای یک سازمان یا جامعه به بار آورد. وجود خطاهای نرم‌افزاری در سیستم‌هایی چون ابزارها و وسایل پزشکی، کنترل بزرگراه‌ها و ترافیک هوایی، خطوط آهن، کنترل دستگاه‌های صنعتی و تجارت الکترونیک می‌تواند فاجعه‌آمیز باشد. علاوه بر این، نفوذ روزافزون تکنولوژی اطلاعات در زندگی روزمره، وجود چنین اشتباهاتی را بیش از پیش غیرقابل تحمل می‌سازد. پیش‌بینی می‌شود که در آینده، مشکل اصلیِ استفاده از تکنولوژی اطلاعاتْ کمبود توان محاسباتی نیست، بلکه عدم توانایی ما در تولید سیستم‌های پیچیده با اطمینان کافی به صحت عملکرد آنهاست.

تحقیقات شرکت IBM در سال ۱۹۹۴ نشان می‌دهد که هزینهٔ اجرای پروژه‌های نرم‌افزاری در ۵۵ درصد موارد بیش از میزان پیش‌بینی شده بوده است؛ اجرای پروژه‌ها به‌طور متوسط در ۶۸ درصد از موارد از زمان پیش‌بینی شده تجاوز کرده و ۸۸ درصد نرم‌افزارها مجدداً طراحی شده‌اند. همچنین اطلاعات منتشره توسط مرکز آمار ایالات متحده در سال ۱۹۹۷ نشان می‌دهد از هر شش سیستم نرم‌افزاری دو سیستم از کار می‌افتند که این میزان برای سیستم‌های بزرگ به ۵۰ درصد می‌رسد. همچنین ۷۵ درصد سیستم‌ها دچار مشکلات عملیاتی هستند.

برخی از خطاهای نرم‌افزاری که منجر به خسارات جدی شده است در اینجا ذکر می‌شود: یک مثال بارز، لغو قرارداد ۸ میلیارد دلاری بین آژانس هوایی فدرال آمریکا و شرکت IBM در سال ۱۹۹۶ برای ساختن نسل جدیدی از سیستم‌های کنترل ترافیک هوایی است. شاید این بزرگترین خسارت در تاریخ مهندسی نرم‌افزار باشد. نمونهٔ دیگر لغو قرارداد بین وزارت دفاع آمریکا و شرکت IBM در سال ۱۹۹۵ برای تولید سیستم‌های اطلاعاتی مدرنی است که قرار بود جایگزین سیستم‌های منسوخ و از کارافتادهٔ قبلی شوند. یک نمونه از مواردی که با شکست گسترده‌ای روبه‌رو شده، نرم‌افزار IBM برای تحویل اطلاعات ورزشی بی‌درنگ (Real time) به سازمان‌ها هنگام بازی‌های المپیک ۱۹۹۶ آتن است. نابودی دو میلیارد دلار در پروژهٔ ماهواره‌ای اروپا و نیز ناکارایی نرم‌افزار سیستم تحویل بار در فرودگاه بین‌المللی دنور که منجر به تأخیر یک و نیم ساله در افتتاح فرودگاه گردید و بابت هر روز تاخیر ۱/۱ میلیون دلار زیان متوجه شرکت هوایی ایالات متحده شد، از جمله نمونه‌های دیگر است.

مهندسی نرم‌افزار چیست؟

تولید یک نرم‌افزار به طور کلی شامل مراحل زیر است:

- تعیین مشخصات، محدودیت‌ها و وظایف سیستم نرم‌افزاری (توصیف نرم‌افزار)
- طراحی نرم‌افزار
- پیاده‌سازی نرم‌افزار
- اعتبارسنجی نرم‌افزار
- نگهداری و تکامل نرم‌افزار

مهندسی نرم‌افزار یک روش مهندسی است که تمام مراحل تولید نرم‌افزار از مراحل ابتدایی تعیین مشخصات سیستم تا نگهداری سیستم بعد از بهره‌برداری را شامل می‌شود. در این تعریف دو عبارت اساسی نهفته است: روش مهندسی و تمام مراحل تولید نرم‌افزار.

- روش مهندسی: مهندسان کارها را به اجرا درمی‌آورند. آنها از نظریه‌ها، روش‌ها و ابزارها در جای مناسب خود استفاده می‌کنند، ولی این نظریه‌ها، روش‌ها و ابزارها را به طور دلخواه انتخاب می‌کنند و همواره سعی دارند تا راه حل مسائل را بیابند، حتی زمانی که هیچ نظریه یا روش کاری موجود نیست که به آن متکی باشند. مهندسین همچنین بر این باورند که باید در محدودهٔ مالی و سیستمی کار کرد تا بتوان راه حل خود را با توجه به واقعیت‌های موجود ارائه کرد.
- مراحل تولید نرم‌افزار: مهندسی نرم‌افزار تنها با فرآیند تخصصی تولید نرم‌افزار درگیر نیست بلکه با فعالیت‌هایی مانند مدیریت پروژه‌های نرم‌افزاری و همچنین توسعهٔ ابزارها، روش‌ها، و نظریه‌ها برای پشتیبانی از تولید نرم‌افزار هم روبه‌روست.

به طور کلی مهندسان نرم‌افزار یک رویکرد سازمان یافته به کار خود دارند. این روش مؤثرترین راه برای تولید نرم‌افزارهای با کیفیت بالاست، با این حال رویکرد خلاقانه و غیر متعارف برای تولید هم ممکن است در بعضی شرایط مؤثر واقع شود. به‌طور کلی، در روش مهندسیْ مؤثرترین شیوه با توجه به مجموعه‌ای از شرایط در نظر گرفته می‌شود.

همچنین باید تصریح کرد که مهندسی نرم‌افزار به معنای خاص آن با علوم کامپیوتر تفاوت دارد. علوم کامپیوتر مربوط به روش‌ها و نظریه‌هایی است که سیستم‌های نرم‌افزاری و کامپیوترها را در برمی‌گیرد در حالی‌که مهندسی نرم‌افزار با مسائل عملی تولید نرم‌افزار روبه‌رو است. البته اطلاع از علوم کامپیوتر برای مهندسان نرم‌افزار ضروری است، همچنانکه آگاهی از فیزیک برای مهندسان الکترونیک ضروری به نظر می‌رسد. ایده‌آل این است که کل مجموعهٔ مهندسی نرم‌افزار را نظریه‌های علوم کامپیوتر دربرداشته و این نظریه‌ها پشتیبان آن باشند، ولی در واقع این طور نیست. مهندسان نرم‌افزار بعضاً از رویکرد موردی برای تولید نرم‌افزار استفاده می‌کنند؛ زیرا هنوز از نظریه‌های علوم کامپیوتر نمی‌توان به‌طور کامل برای مسائل پیچیده و واقعی که نیاز به راه حل نرم‌افزاری دارند استفاده کرد.

روش‌های صوری در مهندسی نرم‌افزار

در دو دههٔ اخیر، روش‌های صوری که روش‌هایی سازمان یافته‌تر با مبنای محکم ریاضی هستند برای تولید و اعتبارسنجی سیستم‌ها مورد توجه قرار گرفته‌اند. معمول است که طراحان سیستم‌های نرم‌افزاری از روش‌های صوری برای توصیف رفتار و ساختار مطلوب و یا اعتبارسنجی سیستم استفاده کنند. با وجود این، هرکس در هر مرحله از تولید نرم افزار می‌تواند از این روش‌ها بهره گیرد. هر روش صوری همچنین باید دارای مجموعه‌ای از راهبردها باشد که شرایط و نحوهٔ بکارگیری آن روش را مشخص کند.

روش صوری می‌تواند هر یک از فعالیت‌های زیر را شامل شود:

۱. توصیف صوری: در این مرحله مشخصات یک سیستم نرم‌افزاری به‌طور دقیق و بی‌ابهام تعیین می‌شود. در توصیف صوری باید محدودیت‌های دامنه‌ای، کارکردی و رفتاری به طور کامل ذکر شوند.

۲. تحلیل صوری: در این مرحله توصیف صوری به دقت تحلیل شده، ناسازگاری‌ها و نقص‌ها به‌طور کامل مشخص می‌شوند.

۳. پالایش: این مرحله حرکت بین سطوح مختلف تجرید، شامل سنجش صحت مدلسازی تا تغییر و پالایش توصیف صوری و پیاده‌سازی مجدد است.

۴. اعتبارسنجی: در این مرحله ویژگی‌های سیستمِ مدلسازی شده به‌دست می‌آید و انطباق آن با توصیف صوری مورد سنجش قرار می‌گیرد.

زبان توصیف صوری ابزاری را برای تعریف دقیق سازگاری، کامل بودن، کارکردها و محدودیت‌ها و در نتیجه پیاده‌سازی دقیق فراهم می‌کند. برای در دست داشتن یک روش اعتبارسنجی صوری باید مدلی برای بیان رفتار سیستم، زبانی برای توصیف خصوصیات مورد نظر سیستم، و دستورالعملی برای تحلیل رفتار سیستم داشت. رفتار سیستم با روشی مبتنی بر ریاضیات بررسی می‌گردد تا از برآورده شدن خصوصیات توصیف شده اطمینان حاصل شود. یکی از روش‌های صوریِ درستی‌یابی نرم‌افزار استفاده از آزمون‌گر مدل است. با در دست داشتن چنین ابزاری قبل از پیاده‌سازی سیستم، می‌توان از صحت مدل آن مطمئن شد. از آنجایی‌که کشف خطاهای منطقی در مراحل اولیهٔ تولید باعث کاهش چشمگیر هزینه‌هاست استفاده از این روش به صرفه است.

روش‌هایی که در مقابل درستی‌یابی صوری قرار دارند، آزمون و شبیه‌سازی است. در آزمون، سیستم واقعی روی ورودی‌های انتخابی اجرا می‌شود و در شبیه‌سازی، مدل شبیه‌سازی شدهٔ سیستم روی ورودی‌های انتخابی (و نه همه آنها) امتحان می‌شود. صوری بودن روش به معنای آن است که ادعای صحت عملکرد سیستم به صورت دقیق ریاضی بیان می‌گردد. در درستی‌یابی باید درستی یا نادرستی ادعای صحت عملکرد اثبات شود. وارسیِ یک نمونه نمایشی از رفتارهای ممکن، مانند آنچه در شبیه‌سازی انجام می‌گیرد کافی نیست بلکه بایستی تضمینی داشت که تمامی رفتارها خصوصیات توصیف شده را در خود دارند.

در پایان باید یادآور شد که استفاده از روش‌های صوری با چالش‌های گوناگونی روبرو است. از جمله، استفاده از روش‌های صوری مستلزم صرف زمان در آغاز چرخهٔ تولید است و لازمهٔ به کارگیری این روش‌ها آشنایی با ریاضیات است که خود مهارت افزون و هزینه‌ای اضافی را می‌طلبد، بدین ترتیب توجیه اقتصادی آن دشوار می‌شود.

ولی با توجه به آنچه گفته شد، برای داشتن نرم‌افزار درست و مطمئن راهی بهتر از استفاده از روش‌های صوری به نظر نمی‌رسد. امروزه استفاده از این روش‌ها به ویژه در طراحی و اعتبارسنجی سیستم‌های بحرانی که هزینهٔ خطا در آنها غیرقابل چشم‌پوشی است رو به افزایش است.

******

منابع:

1. **R.S. Pressman**, *Software Engineering*, 6th edition McGraw Hill, Columbus, 2005.
2. **I. Somerville** , *Software Engineering*, 7th edition Addison Wesley, Boston, 2004.
3. **J.A. Goguen**, *Hidden Algebra for Software Engineering*, in: Combinatorics, Computation and Logic (C. Calude and M. Dinneen, eds.), Proceedings of Conference on Discrete Mathematics and Theoretical Computer Science, 21, Springer, Berlin, 1999, pp. 35-59.
4. **D. Jackson**, *Lectures on Software Engineering*, MIT Press, Cambridge, 2002.
5. **J.M. Wing**, *A Specifier's introduction to formal methods*, IEEE Computer **23** (1990), 8-23.

** **مصطفی زالی و مرجان سیرجانی**، دانشکدهٔ برق و کامپیوتر دانشگاه تهران و پژوهشکدهٔ علوم کامپیوتر پژوهشگاه.